



POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

From Multimedia Data Analysis to the Creation of a Context-aware Framework for the New European Bauhaus

TESI DI LAUREA MAGISTRALE IN
COMPUTER ENGINEERING - INGEGNERIA INFORMATICA

Author: **Tommaso Andaloro**

Student ID: 251837

Advisor: Prof. Letizia Tanca

Co-advisors: Emilia Lenzi, Carlo Andrea Biraghi

Academic Year: 2025-26

Abstract

The New European Bauhaus (NEB) framework provides a comprehensive set of Key Performance Indicators (KPIs) to evaluate urban environments according to the principles of sustainability, aesthetics and inclusion. Although the framework defines a structured methodology for urban assessment, the computation of many indicators still relies on manual procedures, limiting their scalability and applicability to large-scale analyses. This thesis proposes an integrated framework for the automatic computation and contextual interpretation of NEB indicators. Building upon a previous implementation, a fully automated Computer Vision pipeline is developed to compute the existing Beauty indicators B.7 and B.11 directly from street-level imagery. By leveraging state-of-the-art foundation models and object detection architectures, the proposed methodology replaces the manual extraction of architectural features with an automatic workflow, significantly reducing the effort required to evaluate these indicators. Furthermore, the thesis introduces the implementation of the new NEB indicator I.3.7, combining visual information with metadata and structured datasets within a unified computational pipeline. An important contribution of this work is the use of the Context Dimension Tree (CDT), a hierarchical conceptual model designed to represent the possible contexts of a domain of interest, and used here to organise NEB indicators according to contextual information. Through the definition of the appropriate context dimensions and contextual relations, the CDT provides a structured representation of urban context, enabling the integration of visual information, metadata, GIS layers and additional urban datasets into a common analytical framework. The proposed methodology demonstrates how automatic KPI computation and context representation can be combined to support context-aware urban analysis. By integrating heterogeneous sources of information and analyzing the possible contexts in which the data should be considered, the framework provides a scalable foundation for the interpretation of NEB indicators and represents a first step towards more comprehensive decision-support systems for urban design and planning.

Keywords: New European Bauhaus, Urban Context, Context Dimension Tree, Computer Vision, Key Performance Indicators, Urban Analysis

Abstract in lingua italiana

Il framework del New European Bauhaus (NEB) definisce un insieme articolato di Key Performance Indicators (KPI) per la valutazione degli ambienti urbani secondo i principi di sostenibilità, bellezza e inclusione. Sebbene il framework fornisca una metodologia strutturata per l'analisi urbana, il calcolo di numerosi indicatori richiede ancora procedure manuali, limitandone la scalabilità e l'applicabilità a contesti di analisi estesi. Questa tesi propone un framework integrato per il calcolo automatico e l'interpretazione contestuale degli indicatori NEB. Basandosi su una precedente implementazione, viene sviluppata una pipeline completamente automatizzata di Computer Vision per il calcolo degli indicatori Beauty B.7 e B.11 a partire da immagini acquisite a livello stradale. Sfruttando foundation models e architetture di object detection di ultima generazione, la metodologia proposta sostituisce l'estrazione manuale delle caratteristiche architettoniche con un processo completamente automatico, riducendo significativamente il lavoro necessario per il calcolo degli indicatori. Inoltre, la tesi introduce l'implementazione del nuovo indicatore NEB I.3.7, ottenuto attraverso informazioni visive, metadati e dataset strutturati. Il principale contributo del lavoro consiste nell'applicazione del Context Dimension Tree (CDT), un framework gerarchico progettato per rappresentare i vari contesti di un dominio di interesse e utilizzato in questo lavoro per organizzare gli indicatori NEB sulla base delle informazioni contestuali. Attraverso la definizione di context dimensions e context relations, il CDT fornisce una rappresentazione strutturata dei possibili contesti urbani, consentendo l'integrazione di informazioni visive, metadati, layer GIS e ulteriori sorgenti informative all'interno di un unico modello analitico. La metodologia proposta dimostra come il calcolo automatico dei KPI e la rappresentazione del contesto possano essere combinati per supportare analisi urbane context-aware. Integrando sorgenti informative eterogenee e analizzando i possibili contesti nei quali i dati devono essere interpretati, il framework costituisce una base scalabile per l'interpretazione degli indicatori del New European Bauhaus e rappresenta un primo passo verso futuri sistemi di supporto alle decisioni per la progettazione e la pianificazione urbana.

Parole chiave: New European Bauhaus, Context Dimension Tree, Computer Vision,

Contesto urbano, Key Performance Indicators, Analisi urbana

Contents

Abstract	i
Abstract in lingua italiana	iii
Contents	v
1 Introduction	1
2 State of the Art	9
2.1 Technology-Driven Urban Design	9
2.2 Examples of Data-Driven Approaches in Urban Design	9
2.3 The New European Bauhaus	11
2.4 Creating Actionable Future (CrAFt)	14
2.5 Multimedia Data Analysis: A Pipeline for the New European Bauhaus KPIs	15
2.6 Research Gap and Aim of the Thesis	17
2.6.1 How to Measure Beauty	17
2.6.2 How to Support Analysis	19
2.6.3 Context Representation in Urban Design	20
3 Theoretical Background	23
3.1 Computer Vision for Urban Analysis	24
3.2 SAM3: a Foundational Model for Image Segmentation	25
3.3 YOLO Architecture	26
3.3.1 Cross-validation with YOLO	27
3.4 Context Representation through Graph Structures	28
3.5 The Context Dimension Tree	29
4 Automatic Computation of Aesthetic Indicators	33
4.1 Dataset Preparation	33
4.1.1 The dataset	33

4.1.2	Automation of the Process	34
4.1.3	SAM3 for Facade Rectification	35
4.1.4	Manual vs Automated Dataset Preparation	36
4.2	Architectural Elements Recognition	37
4.2.1	Automatic Recognition	37
4.2.2	Manual vs Automated Architectural Elements Recognition	40
4.2.3	Pipeline Efficiency Issues	41
4.3	YOLO for Cross-Validation	41
4.3.1	Data Preparation	42
4.3.2	Implementation	43
4.3.3	Results	44
4.4	Visual Richness Indicator	47
4.5	The Aesthetic Perception Indicator	49
5	The I.3.7 Indicator: Accessibility to Public Transportation	51
5.1	Data Acquisition	52
5.1.1	Metadata from Images	52
5.1.2	Structured Datasets	53
5.2	Accessibility Analysis	54
5.3	Parametrisation	55
5.4	The Model	57
5.4.1	Limits	57
5.4.2	Validation	58
6	A Context-Aware Representation to Support Urban Analysis	61
6.1	The IMM Tree	62
6.2	The NEB Tree	63
6.3	The Context Dimension Tree	64
6.3.1	Content of the implemented Context Dimension Tree	65
6.4	The Pipeline	66
6.4.1	Elaboration of the Input	70
6.4.2	CDT Dimensions Selection	71
6.4.3	CDT Nodes Selection	71
7	Future Developments	75
7.1	The Impact Model	75
7.2	Automatic Integration of User-Provided Datasets	77

8	Conclusions	81
	Bibliography	83
	List of Figures	87

1 | Introduction

With the advent of the digital world in the last few decades, terms like big data and machine learning have become increasingly relevant to every knowledge field, from healthcare to entertainment. In this context, urban and architectural studies make no exceptions, and the importance of interdisciplinary approaches to these branches of knowledge has become more important to grasp the complexity of the world we live in. [1] The ability to access and analyse different datasets and files strengthens the possibility to connect traditional activities in the architectural field, such as buildings' design and environmental analysis, to other related disciplines, such as politics and human rights studies. Moreover, the increasing availability of digital information has profoundly transformed the way urban environments can be observed and interpreted: cities continuously generate large amounts of heterogeneous data, ranging from geospatial information and environmental measurements to photographs, videos, and information produced by public administrations [2, 3]. Each of these sources provides a different perspective on the urban context; therefore, none of them can be considered as sufficient on its own to fully describe the complexity of a city [3]. Nevertheless, despite the growing availability of heterogeneous urban data, the aesthetic quality of the built environment remains difficult to formalise. As Le Corbusier famously described the house as "une machine à habiter"; building upon this idea, this thesis investigates whether at least part of architectural quality can be represented through objective, computable indicators while preserving its contextual dimension.

Addressing this challenge requires more than simply collecting data, but rather concerns the possibility of integrating different datasets and establishing meaningful relationships among them: only through this process, moving from isolated pieces of information towards a broader and more comprehensive understanding of urban phenomena becomes possible [4, 5]. In other words, data acquires its real value when it is analysed together with other datasets, containing additional and enriching sources of knowledge, and further interpreted within a coherent framework, thus supporting a deeper understanding of the context under examination [5].

A fitting example of projects leveraging heterogeneous data to implement a complex

framework supporting spatial and urban analysis is given by Forensic Architecture, a research agency based at Goldsmith, at the University of London: its interdisciplinary team is mainly composed of architects and software developers, and the aim is to analyse multimedia datasets and spatial data to reconstruct incidents of state of violence and human rights violations. Their work demonstrates how heterogeneous data sources can be extracted and correlated to extract important information to interpret complex urban scenarios [6]. In the agency's website's methodology section, for example, many exploited techniques are listed: among them, we can find data mining, software development, machine learning, geolocation data analysis, et cetera. In Figure 1.1 we can see the preview of one of the most important projects of the British team: *A cartography of genocide: Israel's conduct in Gaza since 2023* [7]. The image shows how the project developed an interactive geospatial platform that combines satellite imagery, maps and multiple datasets to document and visualise the spatial distribution of military attacks, damaged infrastructure and evacuation orders across the Gaza Strip.

I.109 A CARTOGRAPHY OF GENOCIDE: ISRAEL'S CONDUCT IN GAZA SINCE OCTOBER 2023

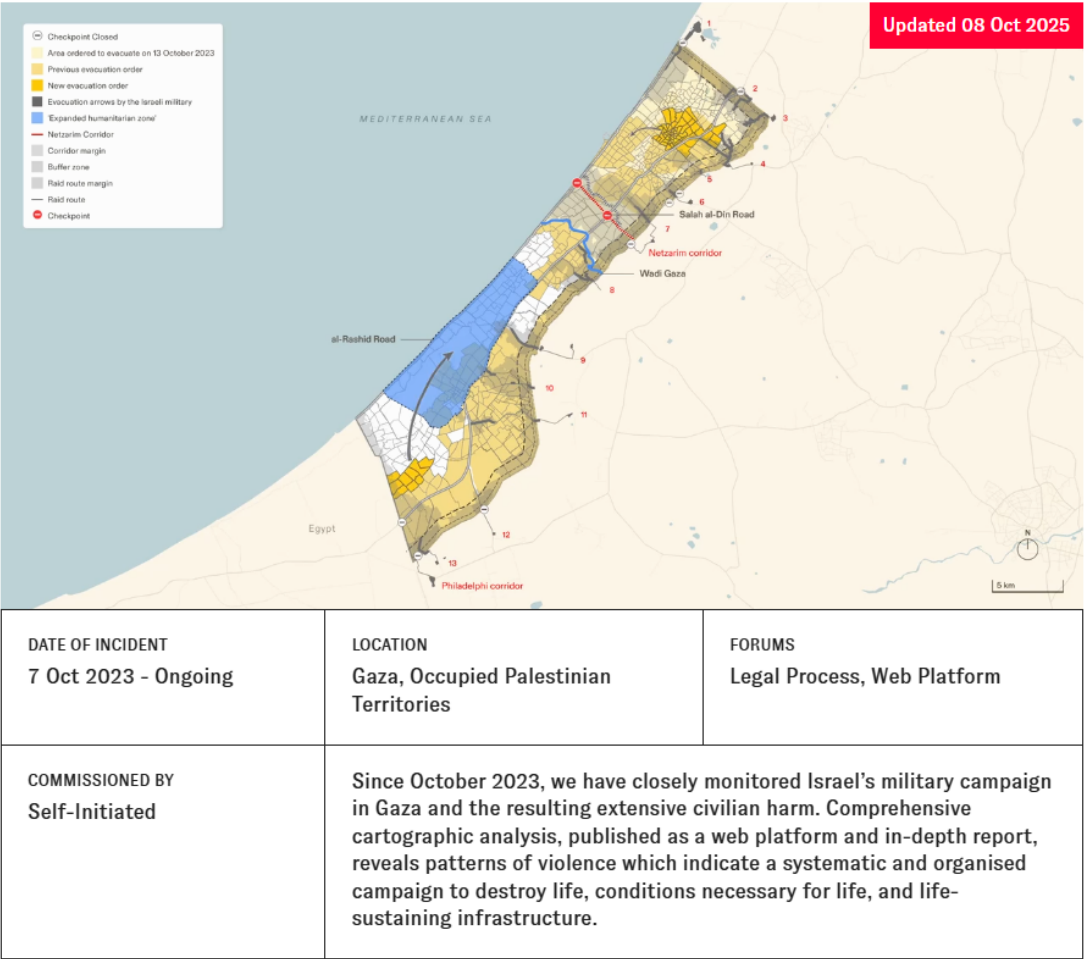


Figure 1.1: Preview of *A cartography of genocide* article.

On the other hand, it is not always clear how to correlate diverse and, in most cases, unstructured data to analyse these scenarios, since their complexity implies that the bare dataset needs to be contextualised to provide insightful and more structured information. In other words, a piece of data may be enriched with further information coming from different sources, or it may be possible to build a semantic model to better understand and fit that information inside a specific analysis [3]. For example, if we have a dataset which contains the bus stops with their relative coordinates, we could integrate this information with another dataset that also contains the timetable for each stop, eventually enhancing the knowledge supporting a potential analysis. For this reason, it is essential to understand the possibilities to analyse different datasets and contextualise them in order to extract qualitative information, which may be realistically used to support urban and architectural studies. Furthermore, extracting information from provided datasets and

linking them to support analytical processes is not sufficient: performance indicators and further information must be contextualised according to a specific scenario, which support a shared and well-understood objective [8]. This means that in architecture and urban studies (as it is in other studies), it is not sufficient to aim at measuring and computing performance indicators to their own ends, but they need to be interpreted following a specific framework or guideline (in this thesis, for example, we will exploit framework and models such as the NEB [9], the CrAFt [10] and the Context Dimension Tree [11]), which has its own objectives and keys of interpretation. If this part is neglected, the outputs may lose their meaning, and they could lead to no conclusion and no actual use [12].

In this context, this thesis aims at applying indicators taken from the New European Bauhaus (NEB) initiative, which promotes solutions that are not only sustainable but also inclusive and beautiful, while respecting the diversity of places, traditions, and cultures in Europe and beyond [13]. These indicators or KPIs are computed exploiting heterogeneous datasets, containing structured, semi-structured, and unstructured data, that, if properly processed, can deliver insightful information which can support urban analysis and design. Leveraging specific tools, such as semantic interpreters, libraries supporting machine learning and other Python-based systems, those indicators will be implemented and will finally contribute to the creation of a framework that maps a contextual view, supporting specific urban analysis tasks. Every step of the project will be supported by two pillar documents: the New European Bauhaus self-assessment tool[9] and the CrAFt Impact Model [10].

The thesis' overview in Figure 1.2 highlights the main steps and objectives of the thesis and proposes a methodology that follows a progressive workflow, where each chapter contributes to the development of the final framework.

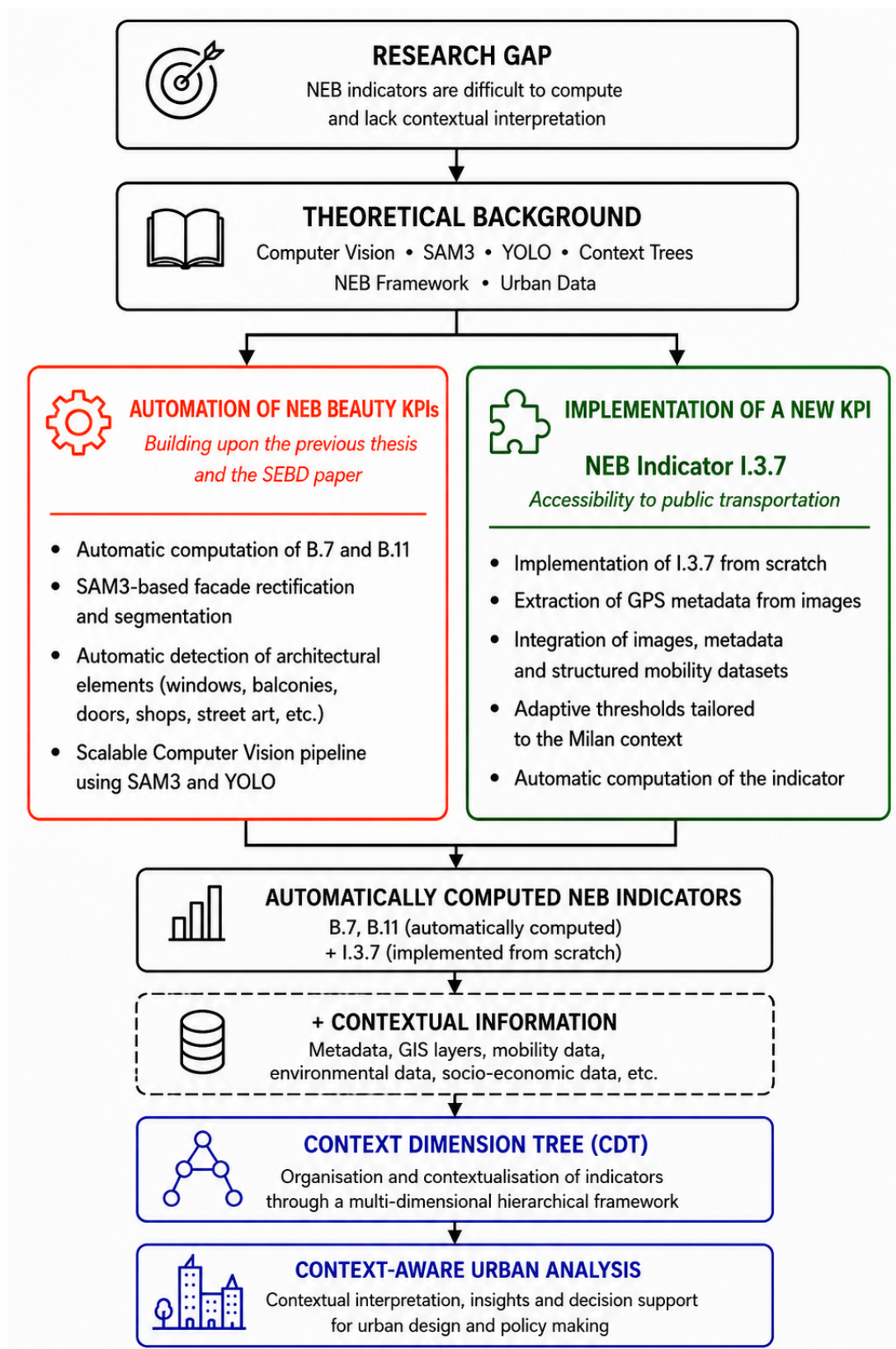


Figure 1.2: Overview of the thesis' work.

Starting from the research gaps discussed above, this thesis provides three main contributions. First, it extends a previous multimedia pipeline by automating the computation of the NEB Beauty indicators B.7 and B.11 through a Computer Vision workflow based on SAM3 and YOLO, considerably reducing the manual effort required by the original implementation. Second, it introduces the implementation of the new Inclusiveness indicator I.3.7, combining image metadata with structured mobility datasets to automatically assess the accessibility of public transportation. Finally, and most importantly, it adopts the Context Dimension Tree (CDT) as a context-aware framework capable of organising heterogeneous urban information and connecting automatically computed indicators with contextual knowledge, supporting a richer interpretation of urban environments.

The thesis is articulated in 8 different chapters, which are organised by objectives and follow the workflow of the given work. The following list describes the overall structure and how every section contributes to the aim of this project:

- **Chapter 2: State of the Art** describes existing literature related to data-driven urban analysis, the New European Bauhaus framework, and context representation. The chapter also discusses the main research gaps addressed by this work, providing the necessary background for the proposed methodology.

This chapter corresponds to the RESEARCH GAP and PREVIOUS THESIS phases of Figure 1.2.

- **Chapter 3: Theoretical Background** presents the theoretical foundations of the proposed framework. In particular, it introduces the Computer Vision models adopted throughout the thesis and discusses graph-based approaches for context representation, leading to the decision to adopt the Context Dimension Tree[11] to represent the possible contexts.

This chapter corresponds to the THEORETICAL BACKGROUND phase of Figure 1.2.

- **Chapter 4: Automatic Computation of Aesthetic Indicators** describes the methodology developed for the automatic computation of two existing NEB Beauty indicators: B.7 and B.11. The chapter presents the complete Computer Vision pipeline, from multimedia data preparation to the computation of the final KPIs, showing how the original manual procedure has been automated through the integration of SAM3 and YOLO. The first section of this chapter is entirely dedicated to multimedia data processing, which differentiate from structured data processing in terms of workflow and techniques. This chapter corresponds to the *Automated*

KPI Computation branch of the proposed framework.

This chapter corresponds to the AUTOMATIC COMPUTATION and IMPLEMENTED NEB INDICATORS phases of Figure 1.2.

- **Chapter 5: The Accessibility to Public Transportation Indicator** presents the implementation of the new NEB indicator I.3.7. The chapter describes the acquisition of the required information, the proposed model and its validation, illustrating how different data sources can be combined to compute a previously unavailable indicator. This chapter corresponds to the *Implementation of a New KPI* branch of the framework.

This chapter corresponds to the IMPLEMENTATION OF A NEW KPI, IMPLEMENTATION FROM SCRATCH and IMPLEMENTED NEB INDICATORS phases of Figure 1.2.

- **Chapter 6: The Context Dimension Tree in Urban Analysis** introduces the proposed Context Dimension Tree and describes its structure and implementation. Starting from the indicators computed in the previous chapters, the framework integrates additional contextual information in order to organise heterogeneous data within a hierarchical representation, enabling context-aware analyses.

This chapter corresponds to the + CONTEXTUAL INFORMATION, CONTEXT DIMENSION TREE and CONTEXT-AWARE URBAN ANALYSIS phases of Figure 1.2.

- **Chapter 7: Future Developments** discusses possible extensions of the proposed framework. In particular, the chapter introduces the concept of an impact model and presents future developments aimed at supporting the automatic integration of user-provided datasets within the Context Dimension Tree.
- **Chapter 8: Conclusions** summarises the main contributions of the thesis and discusses future research directions, highlighting how the proposed methodology combines automatic KPI computation and contextual information to support context-aware urban analysis.

While the automatic computation of NEB indicators represents an important step towards scalable urban analysis, the main contribution of this thesis lies in the adoption of the Context Dimension Tree, which was never used in this setting and provides a unified framework for joining the NEB indicators with heterogeneous contextual information.

2 | State of the Art

2.1. Technology-Driven Urban Design

In the recent decades, urban and architectural studies have been heavily influenced by technology, but lately there has been a shift towards a different approach when it comes to urban design. The increasing availability of environmental spatial data lead researchers to reconsider how to interpret the architectural world, and the growing need to deal with more complex and interconnected built environments is becoming increasingly incumbent. From a technological point of view, in the last years, there has been a shift from traditional urban design, which was based on Computer-Aided Design (CAD) to a more data-driven approach: in particular, we can define “technology-driven urban design” as a design process that combines designers’ creativity with digital computational techniques through rule-based systems involving measurable constraints, parameters, and relations[14]. The previous statement was generic and maybe a bit cliché, but in practice there have been many changes to the techniques and approaches that are used to analyse and design the urban environment, and they can be traced thanks to the numerous projects that have taken place and have been financed by various institutions. The next chapter shows an overview of some of these cases, clarifying how technology and data-driven urban design is changing the traditional approach to urbanism and architecture.

2.2. Examples of Data-Driven Approaches in Urban Design

Many modern and technologically advanced concepts can be applied to the built environment: this section reports three different examples on how data-driven applications support urban analysis and design.

- One of the most discussed approaches is how to integrate IoT systems to urban design and analyses. This case is emblematic for understanding how data can be important to improve the living conditions and ultimately building a better environ-

ment for the inhabitants of urban contexts, since it adds an additional dimension to urban design, which is automatic monitoring of the quantitative performance of the cities' functionality. In Barcelona, for example, which is notable for its high-density population and consequently its high consumption of resources, a new system of smart water networks has been implemented: it consists of a series of sensors which detects leaks in the cities water pipelines. Another example of the application of IoT systems to support urban design comes from Hangzhou, where an extensive network of traffic cameras and street sensors are connected to an artificial 'brain' which leverages AI to solve congestion problems, resulting in CO-2 emissions drops from private transportation, and improving public safety[15].

- Another relevant data-based technology is the digital twin, which found many applications in the urban environment. One representative example is financed by the Municipality of Milan (Figure 2.1), which from 2025 it started developing a digital twin of Milan, which is a digital reconstruction of the city, and it is represented exploiting two different technologies, the point cloud and the 3d mesh. The digital twin constantly updates itself, leveraging the data coming from its physical twin and it is computed to prevent urban related problems and to optimise some efficiencies, all of this is done automatically[16].



Figure 2.1: Milan's digital twin.

- One additional example of data-driven techniques in the design of urban environments is the exploitation of datasets coming from Geographic information Systems (GIS), which are integrated to computer vision systems to deliver high value rep-

representations of the cities. In this perspective, the European Space Agency (ESA) implemented a complex network of space radars, which collects important data concerning diverse dimension of the built environment. An interesting application of this is the juxtaposition of different reconstructions of the city centre of Venice through SAR interferometry: this technique maps the problematic areas which are sinking more rapidly in the lagoon[17].

Although these approaches differ significantly in terms of objectives and technologies, they all demonstrate a common trend: urban environments are increasingly interpreted through heterogeneous digital datasets, whose integration has become a key challenge for contemporary urban analysis.

2.3. The New European Bauhaus

In the last decade, European citizens have become increasingly aware of the threat caused by climate change and environmental degradation, and this led to a complete revolution in terms of perception of our life and of everyone's impact on the health of our planet. The European Green Deal is probably the most emblematic attempt to change Europe's growth strategy, and it aims at transforming the Union into a modern, resource-efficient and competitive economy. Under this light, a new initiative has taken afoot, which seeks to play a critical role in enabling the green transition of the European society, connecting the European Green Deal to buildings, living spaces and experiences: the New European Bauhaus (NEB)[9], which logo and poster are shown in Figure 2.2.

Buildings, from their construction to their usability, are major energy consumers and a significant source of damaging emissions. Moreover, the urban context of the European Union is the product of centuries of architectural progress and research, resulting in a strong heritage which must be preserved and which imposes an elaborated culture in the design of new buildings. Finally, the relation between new architectures and citizen must be taken into consideration, since it sparks complex dynamics which impact on the economic side of people's lives. In this perspective, the NEB tries to assess the quality and the impact of three pillars which are related to the built environment: Sustainability, Beauty and Inclusiveness. In particular:

- Sustainability: the ability of buildings and urban environments to minimise their environmental impact, reduce resource consumption and contribute to climate neutrality, while ensuring resilience against future environmental challenges. ;
- Beauty: the quality of spaces, architectures and experiences in terms of aesthetics,

cultural value and emotional impact, promoting environments that are pleasant, meaningful and capable of strengthening the relationship between people and places.

- **Inclusiveness:** the capacity of the built environment to be accessible, affordable and welcoming for all citizens, fostering social cohesion, participation and equal opportunities regardless of age, gender, economic condition or physical ability.

In order to translate these principles into measurable objectives, the European Commission proposed a framework composed of indicators and Key Performance Indicators (KPIs), aimed at evaluating the contribution of buildings and urban environments to the goals of the New European Bauhaus. The framework covers a broad range of aspects, spanning from environmental sustainability and resource efficiency to social accessibility, quality of public spaces and architectural value, making the assessment process particularly complex and dependent on information coming from different domains. As a consequence, the computation of such indicators requires the integration of heterogeneous sources of knowledge, capable of describing both the physical characteristics of the built environment and the way citizens interact with it.



Figure 2.2: The New European Bauhaus logo and promotional poster.

At the same time, the increasing availability of digital data has opened new possibilities for urban analysis and decision-making processes. Geographic information systems, open urban datasets, remote sensing technologies and multimedia repositories continuously generate large amounts of information describing cities and buildings from different per-

spectives, creating the conditions for the development of data-driven methodologies able to support urban design and policy making. In this context, artificial intelligence and computer vision techniques have progressively gained relevance, allowing researchers to automatically extract meaningful information from urban data and to analyse phenomena that would otherwise require extensive manual surveys.

The New European Bauhaus represents a particularly interesting application scenario for these approaches, since many of its indicators involve dimensions that are inherently difficult to quantify and often require the combination of multiple sources of information. For this reason, several research activities have focused on the development of methodologies capable of transforming heterogeneous urban data into quantitative measures aligned with the NEB framework, with the objective of supporting both the monitoring of existing urban realities and the design of future interventions, as shown in Figure 2.3. The following sections present an overview of the main contributions available in the literature and of the approaches that have been proposed to support the computation of NEB indicators. Among the numerous KPIs proposed by the NEB framework, this thesis focuses on three specific ones: the Beauty indicators B.7 and B.11 evaluate architectural qualities of building façades, such as compositional order and chromatic harmony, making them particularly suitable for image-based analysis. In addition, the Inclusiveness indicator I.3.7 measures the accessibility of public transportation, requiring the integration of spatial and metadata-based information. These three indicators have been selected because they allow the combination of multimedia analysis, structured data processing and contextual information within a common computational framework.

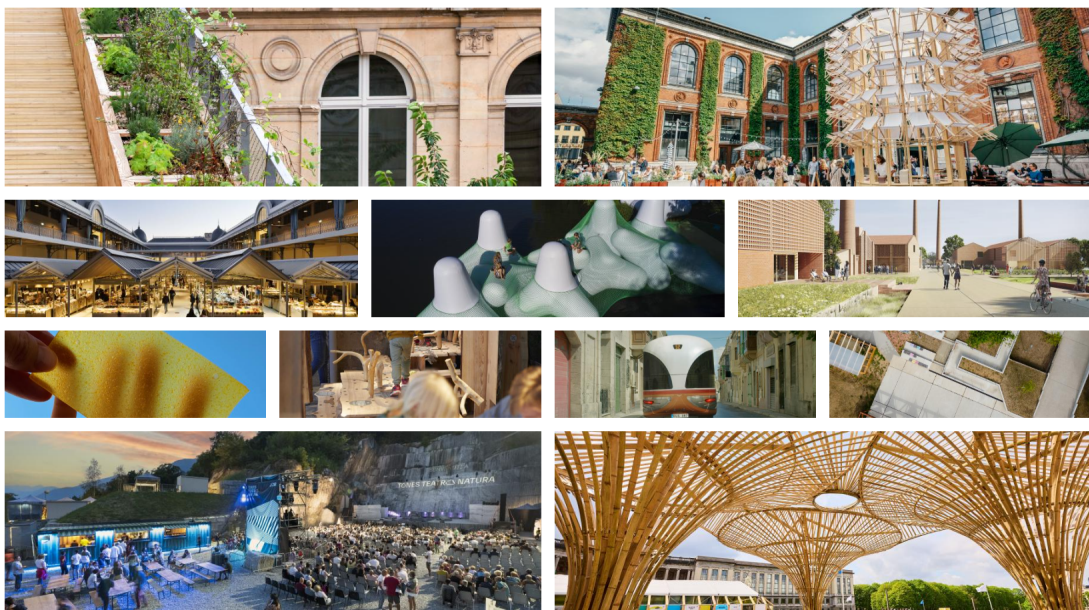


Figure 2.3: Image taken from NEB official website.

2.4. Creating Actionable Future (CrAFt)

While the New European Bauhaus provides a conceptual framework capable of describing what a sustainable, beautiful and inclusive urban environment should be, the practical assessment of these dimensions remains a complex challenge. In particular, many of the values promoted by the NEB are inherently multidimensional and often difficult to quantify through traditional urban indicators. For this reason, several European research initiatives have focused on the development of methodologies capable of translating these principles into operational assessment frameworks.

One of the most relevant examples is represented by the CrAFt project (Creating Actionable Futures) [10], a Horizon Europe initiative aimed at supporting cities in their transition towards climate neutrality while maintaining a strong alignment with the values promoted by the New European Bauhaus. Within the project, a specific framework called the NEB Impact Model has been developed with the objective of supporting urban administrations in the monitoring and evaluation of complex urban interventions. Rather than being conceived as a rigid indicator system, the model is intended as a flexible and context-sensitive tool, capable of integrating existing assessment methodologies and adapting them to the specific characteristics of different urban realities. A key aspect of the CrAFt approach is the adoption of a systemic perspective. The model assumes that urban interventions rarely generate isolated effects; on the contrary, actions performed in one domain often produce consequences across multiple dimensions of urban life. For this reason, the framework is designed not only to assess direct impacts, but also to identify co-benefits and interactions among different aspects of sustainability, inclusiveness and beauty. Examples discussed within the project include building retrofitting interventions, sustainable mobility transformations and renewable energy infrastructures, all of which are analysed through networks of interconnected impacts rather than through isolated indicators. The structure of the NEB Impact Model is organised around five main pillars: governance, economic performance, socio-cultural performance, healthy living and environmental performance. These pillars are further articulated into seventeen impact categories and a set of suggested indicators, as shown in Figure 2.4, spanning from climate neutrality and sustainable mobility to cultural sustainability, affordability, participation and quality of public spaces. Such a structure reflects the intrinsic complexity of contemporary urban systems and highlights the necessity of combining heterogeneous sources of information in order to obtain a meaningful representation of urban quality.

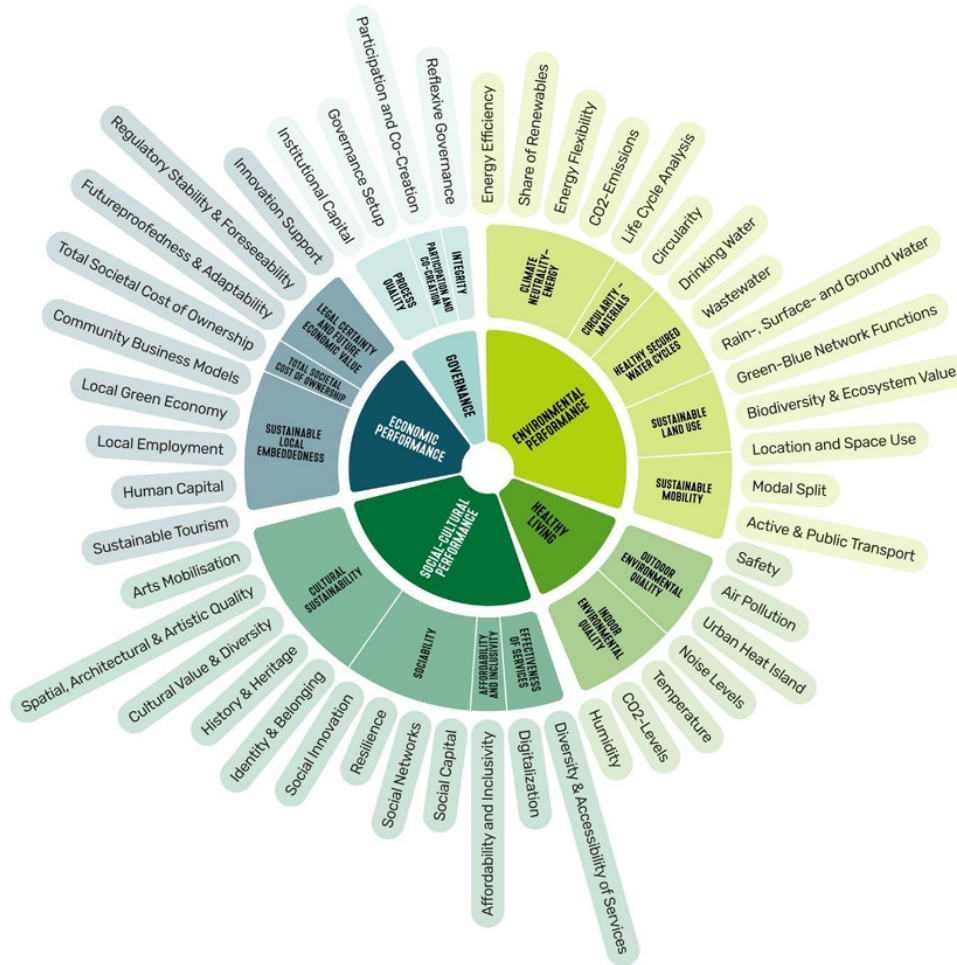


Figure 2.4: The CrAFt framework.

The relevance of the CrAFt project for this thesis lies precisely in this attempt to operationalise the principles of the New European Bauhaus through measurable indicators. In particular, the work presented in the following chapters contributes to this broader research direction by investigating how multimedia data, computer vision techniques and contextual information can support the computation of specific NEB indicators, reducing the need for manual assessments and enabling scalable analyses of the built environment.

2.5. Multimedia Data Analysis: A Pipeline for the New European Bauhaus KPIs

This thesis builds upon previous research that attempted to measure some of the Beauty-related indicators proposed within the New European Bauhaus framework. The methodology was initially developed during a previous thesis project [18] and was subsequently

consolidated in the discussion paper presented at SEBD 2026 [19]. Starting from a dataset of images collected in the Lambrate district of Milan, the previous work developed a computational pipeline capable of transforming visual architectural characteristics into measurable quantities. Since the available information was limited to street-level photographs, the analysis focused on the KPIs B.7, B.10 and B.11; these indicators were considered the most suitable for an image-based interpretation of architectural quality and harmony. The implemented pipeline (Figure 2.5) consisted of five main stages: data acquisition, image cleaning, feature extraction, KPI computation and validation. A significant effort was dedicated to the preparation of the dataset; images were geometrically corrected to reduce perspective distortions and manually annotated through Roboflow in order to identify architectural elements such as windows, balconies, entrance doors, shops and street art. Preliminary experiments with YOLO-based object detection were also conducted; however, due to dataset limitations and scene complexity, the final workflow still relied mainly on manual annotations. Starting from the extracted annotations, several visual characteristics were computed: symmetry was used as an approximation of compositional order; the ratio between openings and surface contributed to the evaluation of contrast; dominant colours were extracted through clustering techniques; finally, geometric operations were employed to analyse the overall shape of the facade. These features were then combined to approximate the selected NEB indicators, with KPI B.10 representing a synthetic measure of the relationship between the building and its surrounding context. Validation results showed that the methodology was capable of capturing measurable differences between buildings belonging to different urban environments.

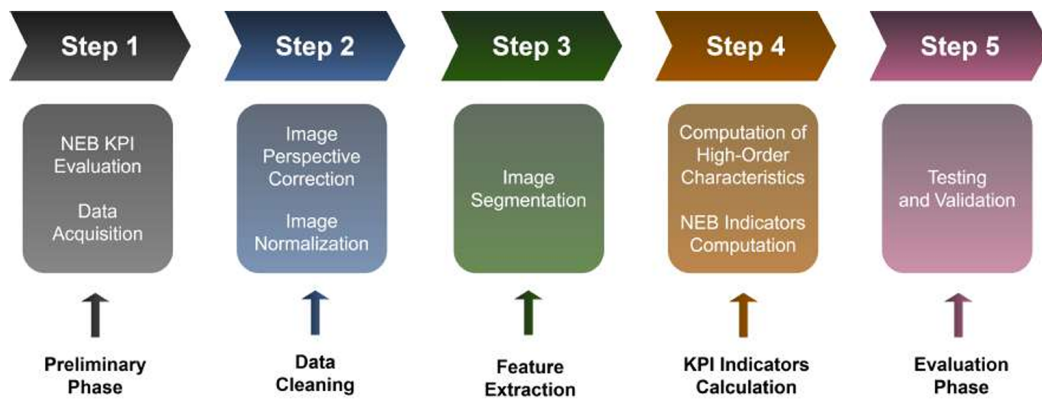


Figure 2.5: The pipeline of the previous thesis.

Despite these encouraging results, the previous research still presented several limitations: many operations required significant manual intervention, some indicators had to be strongly simplified, and the notion of context remained largely based on visual com-

parisons between facades. For this reason, the present thesis starts from the foundations established by the previous work and extends them towards a broader and more context-aware perspective; buildings are no longer analysed as isolated entities, but as components of a wider urban structure where architectural elements, metadata and spatial relationships contribute to the interpretation of the surrounding environment.

2.6. Research Gap and Aim of the Thesis

The New European Bauhaus provides a complex and detailed framework to analyse and describe urban environments in a deterministic way, and on the other hand several research challenges emerge, e.g.: how beauty can be measured in a systematic way, how digital technologies can support urban analysis and indicator extraction, and how contextual information can be represented in a coherent framework capable of supporting decision-making processes. In fact, the NEB fails to address various concerns related to qualitative and subjective indicators – such as many of the KPIs belonging to the Beauty dimension – and no conceptual framework capable of supporting the practical application of its categories to the physical environment exist. In other words, the initiative offers a robust structure for assessment and evaluation, but does not fully explain how such indicators can be operationalised, measured and connected to real-world urban data. This limitation becomes particularly evident when analysing urban environments through digital tools: while quantitative indicators can often be computed through existing datasets and statistical methods, qualitative dimensions require *the integration of heterogeneous information sources, contextual interpretation and advanced computational techniques*.

The aim of this thesis is to investigate these challenges and to propose a methodology which combines computer vision, urban data analysis and contextual modelling, thus creating a framework capable of supporting the evaluation of selected New European Bauhaus indicators while preserving the complexity of urban environments.

2.6.1. How to Measure Beauty

As mentioned before, one of the aims of the thesis is to try and evaluate beauty indicators, which are related to urban design and analysis. There has been much research in this field which achieved important results and which leveraged innovative data-driven techniques to try and formalise aesthetic indicators. A well-fitting example is given by researchers from King’s College London, in collaboration with Nokia Bell Labs, who developed an automatic tool named *Facelift*[20]. The system not only exploits deep learning techniques to understand whether an urban context can be considered aesthetically pleasing, but

is also capable of "beautifying" a given urban scene and identifying the characterising elements of this beautified version, as described in Figure 2.6. The problem raised by this kind of study is that machine learning and deep learning techniques are extremely effective when it comes to classification tasks; however, their inner "black-box" nature makes it difficult to understand which metrics the model actually used and which visual elements contributed to the final judgement.

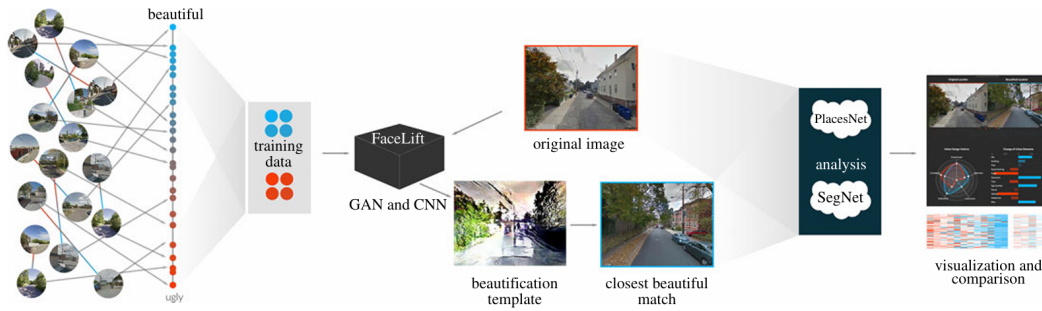


Figure 2.6: The Facelift framework.

One of the most interesting aspects of this approach is that it makes it possible to analyse entire cities at a scale which would be impossible through traditional manual surveys. Moreover, the information extracted can support urban planners and designers in identifying areas that may require aesthetic improvements, while also understanding which visual characteristics positively influence the perception of urban quality. This demonstrates how multimedia data analysis and artificial intelligence can support urban design not only from a functional point of view, but also from a aesthetic and perceptual perspective.

Despite these advancements, a fundamental issue remains unresolved: beauty is not solely determined by measurable visual features. The architectural perception is influenced by cultural, historical, and social factors, which are often difficult to encode into computational models. Consequently, current approaches can successfully identify correlations between visual characteristics and aesthetic preferences; nevertheless, they still struggle to provide a comprehensive explanation of why a specific urban environment is perceived as beautiful. This limitation highlights one of the central research gaps addressed in this thesis: the need to integrate quantitative visual indicators with a contextual information capable of enriching their interpretation.

2.6.2. How to Support Analysis

At this point, an emblematic question may arise: is it possible to standardise and translate into "bits" the whole process of urban design from an aesthetic point of view? In other words, is it possible to map and structure qualitative and aesthetic factors characterising an architectural environment? This thesis starts from images of buildings and urban environments and tries to compute a subset of the KPIs formalised within the New European Bauhaus framework, thus supporting urban analysis through digital methodologies. Such an objective naturally leads to the adoption of computer-vision techniques, which have recently become one of the most promising tools for extracting meaningful information from visual data. Le Corbusier famously stated that "Une maison est une machine à habiter"; such a statement reflects a rationalist vision of architecture, where buildings can be understood through objective principles and functional relationships. While contemporary urban environments are undoubtedly more complex than the mechanistic vision proposed by early modernism, the increasing availability of digital data and computational tools suggests that at least part of the knowledge traditionally employed by planners and designers can be formalised and analysed automatically.

Recent advances in computer vision have demonstrated that urban images contain a remarkable amount of information which can be extracted through machine learning techniques. Tasks such as object detection, semantic segmentation and scene understanding make it possible to identify buildings, vegetation, transportation infrastructures, public spaces and other urban elements directly from photographs or street-view imagery. Consequently, images can be transformed into structured datasets which support quantitative analysis. On the other hand, when aesthetics are involved, researchers are still debating whether taste can be fully formalised into a structured computational model. Recent studies demonstrated that architectural stimuli elicit significantly more variable responses than natural landscapes or human faces, which generally produce more consistent preference patterns. Furthermore, architectural judgement is deeply influenced by cultural background, personal experiences and local traditions[21]. These findings suggest that it may be difficult to identify a universal model capable of assigning an absolute aesthetic value to a single building. However, they also suggest that beauty can be investigated from a relational perspective: while the aesthetic quality of an isolated object may be difficult to formalise, its coherence, harmony and consistency with the surrounding urban context can be analysed through measurable indicators. Therefore, absolute indicators may be insufficient to represent architectural beauty; conversely, contextual and relative indicators may provide a more robust framework for aesthetic assessment. This consideration introduces a second research gap: although computer vision techniques are increasingly

capable of extracting information from urban environments, they often lack a mechanism capable of contextualising such information. As a result, the extracted data remain isolated observations, while urban analysis requires the ability to understand relationships, dependencies and interactions among different elements of the city.

2.6.3. Context Representation in Urban Design

The complexity of urban environments does not emerge solely from the quantity of available information, but also from the intricate network of relationships among different urban elements. Buildings, public spaces, mobility systems, environmental factors and social dynamics continuously interact with each other; consequently, analysing a single component without considering its context often leads to incomplete interpretations. For this reason, context representation has progressively become a central topic within urban informatics, urban digital twins and context-aware computing. The fundamental idea behind these approaches is that information should not be considered as isolated data points, but rather as entities embedded within a broader semantic structure capable of describing their relationships and meaning. Several authors proposed ontology-based models, semantic networks and hierarchical representations to organise urban knowledge and support decision-making processes. These frameworks make it possible to integrate heterogeneous datasets while preserving their semantic relationships; thus facilitating data exploration, analysis and interpretation. Nevertheless, many existing solutions focus primarily on quantitative information and often struggle to integrate qualitative dimensions such as perception, beauty and urban experience[22]. This limitation becomes particularly relevant when considering the New European Bauhaus framework. The initiative defines a rich set of interconnected indicators, but it does not provide a representation model capable of describing how these indicators relate to one another, how they influence the perception of urban quality, or how they should be connected to the underlying data sources.

Consequently, a third research gap emerges: the necessity of developing a contextual representation capable of connecting urban data, indicators and qualitative assessments within a coherent structure. The hypothesis explored in this thesis is that a hierarchical context-based model can provide such support; allowing information extracted from heterogeneous sources to be organised, contextualised and subsequently used to improve the computation and interpretation of New European Bauhaus indicators. The Context Dimension Tree (CDT) [11], described in the following chapters, was developed precisely to address this challenge, and will be adopted in this work. By representing urban knowledge through a hierarchical structure composed of interconnected contextual dimensions,

it becomes possible to: enrich data interpretation, support indicator computation, and establish meaningful relationships between quantitative measurements and qualitative urban characteristics. A structured representation of the research gaps mentioned above and the proposed framework are reported in Figure 2.7

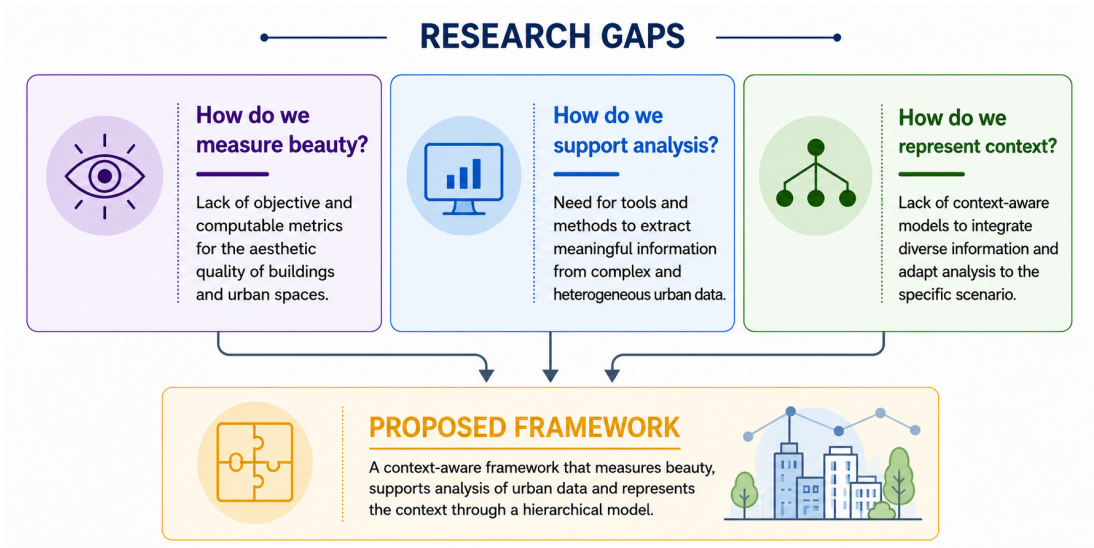


Figure 2.7: Research gaps and aim of the thesis framework.

3 | Theoretical Background

The methodological framework adopted in this research is grounded in recent developments within the fields of computer vision and deep learning; in particular, the study relies on techniques for image understanding that enable the automatic identification and characterization of elements contained in urban environments. Over the last decade, the increasing availability of large-scale visual datasets, together with the rapid evolution of neural network architectures, has substantially transformed the way urban spaces can be analysed; tasks that traditionally required extensive manual annotation and observation can now be performed through automated systems capable of extracting semantic information directly from images.

Within this context, urban analysis has progressively incorporated computer-vision methodologies as a means of quantifying and describing the built environment; object detection and image segmentation models, in particular, have become fundamental tools for identifying the physical components that contribute to the visual and functional characteristics of urban spaces. However, the extraction of information alone is not sufficient to fully describe the complexity of urban environments; the detected elements must also be organised within a framework capable of representing the contextual relationships that characterize a specific place. For this reason, alongside the computer vision methodologies adopted for data extraction, this research also relies on a model for contextual representation based on graph structures and on the Context Dimension Tree (CDT), which provides the conceptual foundation for organising and interpreting urban information.

The following sections therefore provide an overview of the theoretical concepts and architectures that constitute the basis of this work; first, the role of computer vision in urban analysis is discussed, and then the Segment Anything Model (SAM) is introduced as a segmentation framework for the extraction of urban elements, followed by the YOLO architecture for object detection. Finally, the principles of context representation through graph structures and the Context Dimension Tree are presented as the conceptual framework used to organise and contextualise the extracted information.

3.1. Computer Vision for Urban Analysis

Computer vision is a branch of artificial intelligence concerned with the automatic extraction of information from visual data; its objective is to transform images into structured representations that can be interpreted, analysed and used to support decision-making processes. Recent advances in deep learning have significantly expanded the capabilities of computer vision systems; tasks that previously required handcrafted features and extensive manual intervention can now be performed through neural networks capable of learning complex visual patterns directly from large image datasets. The growing availability of visual data sources has contributed to the widespread adoption of computer vision techniques within urban studies: street-level imagery, satellite images, drones, and surveillance systems continuously generate large quantities of information describing the physical characteristics of cities. Such datasets provide an unprecedented opportunity to observe urban environments at different spatial scales; however, their size makes traditional manual analysis increasingly impractical. Within this context, computer vision has emerged as an effective tool for supporting urban analysis. Through techniques such as image classification, object detection, semantic segmentation and instance segmentation, it is possible to automatically identify buildings, roads, vegetation, vehicles, pedestrians and many other components of the built environment. The extracted information can then be employed to analyse urban morphology, monitor environmental conditions, evaluate mobility patterns, assess public spaces and support planning activities.

Recent literature demonstrates how computer vision applications have progressively expanded beyond the simple recognition of physical objects; visual data are increasingly used to derive indicators related to urban vitality, walkability, environmental quality, safety, public health and even human perception of urban spaces[23]. As a consequence, computer vision is no longer limited to the description of urban form alone; rather, it represents a bridge between the physical characteristics of the built environment and the social, environmental and perceptual dimensions that define the quality of urban life. For these reasons, computer vision constitutes a fundamental component of the methodology adopted in this research; the information extracted from street-level imagery provides the basis for the identification of urban elements and for the subsequent construction of contextual representations aimed at supporting the evaluation of urban environments.

3.2. SAM3: a Foundational Model for Image Segmentation

The Segment Anything Model 3 (SAM3), introduced by Meta AI, represents one of the most significant advances in image segmentation achieved in recent years; unlike conventional segmentation models, which are generally trained to recognise a predefined set of object categories, SAM3 was designed as a foundation model capable of producing segmentation masks for a potentially unlimited variety of visual entities. This characteristic distinguishes SAM3 from previous approaches; rather than focusing on category-specific recognition, the model aims to identify and delineate any visually coherent object present within an image[24]. From an architectural perspective, SAM3 consists of three principal components: an image encoder, a prompt encoder, and a mask decoder.

1. The image encoder is responsible for generating a high-dimensional representation of the visual content contained in the image;
2. The prompt encoder processes additional inputs, such as points, bounding boxes, or masks, which provide guidance regarding the target object;
3. The mask decoder combines the information produced by the previous modules and generates the final segmentation output.

Through this design, the model is able to achieve high levels of flexibility while maintaining strong generalisation capabilities across different domains. The relevance of SAM3 for urban analysis derives primarily from its ability to accurately delineate objects without requiring extensive task-specific retraining; urban environments are characterised by a high degree of visual complexity, since multiple categories of elements coexist within the same scene and often exhibit substantial variability in terms of scale, shape, and appearance. Under these conditions, traditional segmentation approaches may require large quantities of labelled data in order to achieve satisfactory performance; SAM3, by contrast, can leverage its pre-trained knowledge to identify and segment a wide range of urban components with limited additional supervision.

Within the context of this research, SAM is employed as a tool for the identification and segmentation of traditional urban elements (Figure 3.1): the generated masks provide detailed spatial information regarding the location and extent of objects within the scene, thereby enabling a more comprehensive representation of the urban environment. Furthermore, segmentation outputs can be integrated with object detection results in order to obtain complementary information; while object detectors primarily focus on recognising

and localising specific categories, segmentation models provide a pixel-level description of the visual content, allowing for a more detailed understanding of urban scenes.



Figure 3.1: Example of building segmentation in an urban landscape.

3.3. YOLO Architecture

Among the various object detection models developed in recent years, YOLO (You Only Look Once) has become one of the most widely adopted solutions; its popularity derives from the ability to combine detection accuracy and computational efficiency, making it particularly suitable for applications involving the analysis of large quantities of images. Since its first introduction, the architecture has undergone several revisions and improvements; nevertheless, the fundamental idea behind the model has remained substantially unchanged: object detection is treated as a single task, where the localisation and classification of objects are performed simultaneously through a single pass of the neural network. This approach differs significantly from traditional two-stage detectors, such as the R-CNN family of models; in those architectures, the detection process is divided into separate phases, where candidate regions are first generated and subsequently classified. While this strategy often produces highly accurate results, it also introduces additional computational costs and longer inference times; YOLO addresses these limitations by integrating the entire detection process into a unified architecture, allowing images to be processed much more efficiently while maintaining competitive performance[25].

From a structural perspective, modern versions of YOLO are generally composed of three main components, as shown in Figure 3.2

1. A **backbone**, responsible for extracting visual features from the input image;
2. A **neck**, which combines information at different scales and improves the detection of objects characterised by different dimensions;
3. A **detection head**, which produces the final predictions in terms of object location, confidence score and class.

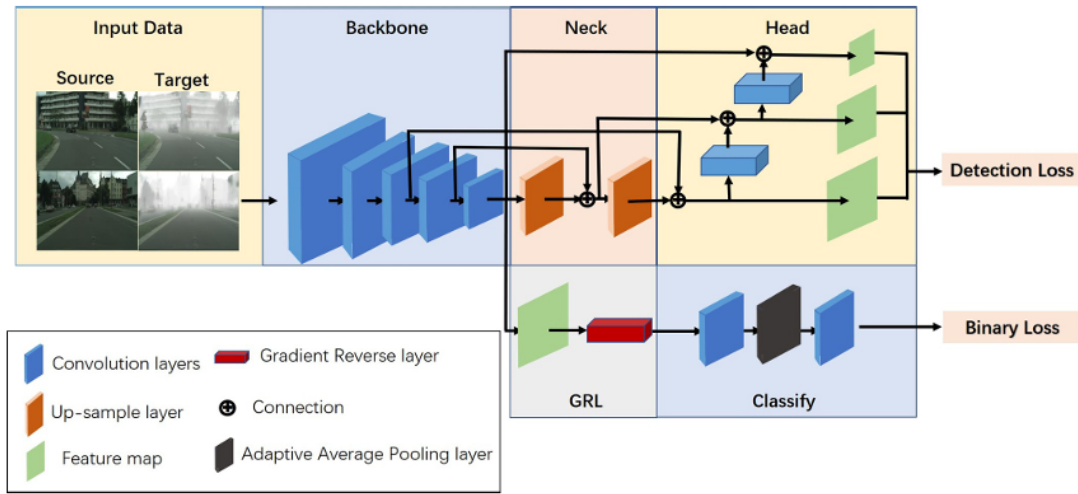


Figure 3.2: YOLO's structure framework.

This multi-scale approach is particularly relevant in urban environments, where objects may vary considerably in size, appearance and spatial distribution.

The adoption of YOLO within urban analysis has increased significantly over the last years; the model has been employed to identify vehicles, pedestrians, street furniture, traffic signs, vegetation and many other elements of the built environment. Such capability is particularly valuable when dealing with extensive street-level imagery datasets, where manual annotation would be impractical; by automatically extracting semantic information from urban scenes, object detection models make it possible to transform visual observations into structured data that can subsequently support urban analysis and decision-making processes.

Within this research, YOLO is employed as the primary framework for the identification of urban elements contained in street-level imagery; the detected objects constitute one of the main sources of information used to characterise the observed environment and, together with the segmentation results, contribute to the construction of a richer representation of the urban context.

3.3.1. Cross-validation with YOLO

One of the main application of the YOLO library is cross-validation: the main idea is to divide the dataset into a number of subsets, called folds and at each iteration, one fold is used for validation while the remaining folds are used for training. The process is repeated until each fold has been used as validation data exactly once. In this way, every sample contributes both to the training and to the evaluation phases, reducing the risk that the

model performance is excessively influenced by a particular partition of the dataset. The final performance of the model is then obtained by averaging the results collected across all the iterations; this provides a more reliable estimate of its generalization capability and allows a more efficient use of the available data[26]. In the Figure 3.3, the process of cross-validation is outlined.

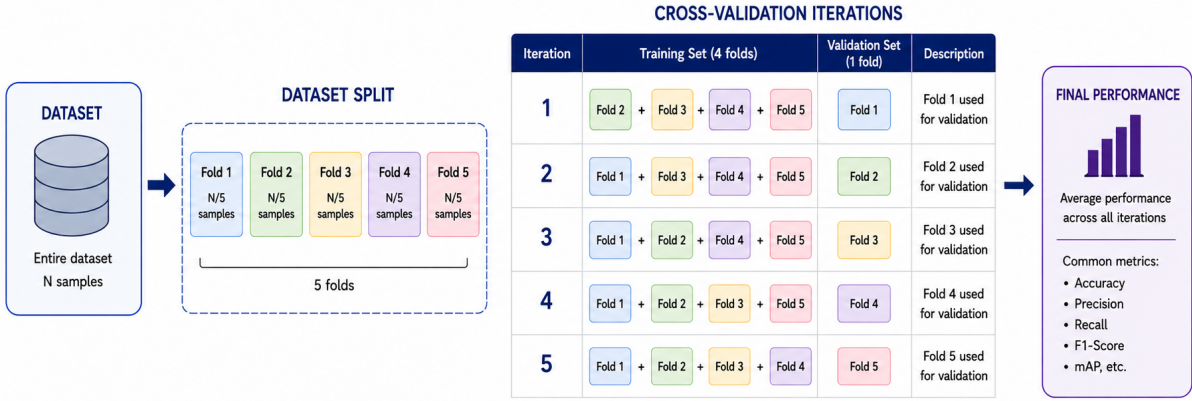


Figure 3.3: Cross validation process.

3.4. Context Representation through Graph Structures

The representation of context constitutes a fundamental problem in context-aware systems; before a system can adapt its behaviour to a specific situation, it must be provided with a formal model capable of describing the relevant characteristics of the environment in which it operates. While the notion of context has been interpreted in different ways across the literature, a common requirement concerns the ability to capture not only individual pieces of information, but also the relationships that exist among them; context, in fact, is rarely defined by a single variable, and is more frequently the result of the interaction among multiple dimensions describing users, objects, locations, activities, temporal conditions, and environmental characteristics. Within this perspective, graph-based representations have emerged as particularly effective modelling tools; by representing entities as nodes and semantic relationships as edges, graph structures make it possible to capture both the individual components of a context and the dependencies that connect them. Unlike flat representations, where contextual information is stored

as independent attributes, graph models allow the explicit representation of hierarchical, associative, and dependency relationships; consequently, they provide a richer and more flexible description of complex environments.

The adoption of graph structures is especially relevant when dealing with dynamic and heterogeneous domains, such as urban environments, where contextual information originates from multiple sources and evolves continuously over time. In such scenarios, the contextual meaning of an observation often depends on additional information describing the surrounding situation; therefore, the representation model must be capable of expressing different levels of abstraction while preserving semantic relationships among contextual elements. Among the various graph-based approaches proposed in the literature, hierarchical models occupy a particularly important position; their objective is not merely to store contextual information, but rather to organise it according to conceptual dimensions that reflect the structure of the observed domain. In this way, context can be represented as the combination of multiple perspectives, each contributing to the characterization of a specific situation. Such an approach enables the identification of contextual configurations through the selection of values along different dimensions; the resulting representation is both semantically meaningful and computationally manageable. The framework adopted in this work follows this line of research and relies on a hierarchical graph representation known as the Context Dimension Tree (CDT), defined in the next section: rather than describing context as an unstructured collection of attributes, the CDT models it as a structured combination of dimensions, concepts, and relationships. This representation provides the conceptual foundation upon which context-aware reasoning can be performed; furthermore, it allows contextual information to be explicitly connected to the analytical processes operating on the data.

3.5. The Context Dimension Tree

The Context Dimension Tree (CDT) was introduced as a conceptual model for representing context in a structured and semantically meaningful manner [11]. In this thesis, its objective is not to represent the whole urban environment, but rather the different contexts that may arise within it, by organising contextual information according to a hierarchy of dimensions and concepts. More generally, the CDT aims to characterise the situation of an entity – where an entity is defined as a person, place, or object that is relevant to the interaction between a user and an application – in different application contexts [11].

The CDT is used to describe complex situations through multiple contextual perspec-

tives. Each perspective corresponds to a context dimension, and the combination of the values assumed by the different dimensions defines the current context. Such contextual information can then be exploited to refine a query or to support context-aware analysis according to the specific application.

From a structural point of view, the CDT is a tree composed of four different types of nodes:

- **Root node:** it is a star-shaped white node that represents the origin of the tree.
- **Dimension node:** it is a black circular node representing a context dimension, i.e., one of the relevant perspectives through which the analysed situation can be described.
- **Concept node:** each dimension node has one or more white circular children representing the possible values that the dimension can assume. Concept nodes may in turn contain additional (sub-)dimension nodes, thus generating a recursive hierarchical structure.
- **Attribute node:** it is a leaf of the tree represented as a black square node. If attached to a dimension node, the attribute acts as a shorthand when the number of possible concepts is particularly large. If attached to a concept node, it represents one of its possible values.

For example, in a given environment, a dimension describing the role of a person may contain concepts such as *farmer*, *oenologist*, or *driver*; similarly, a temporal dimension may contain concepts describing different operational phases, as shown in Figure 3.4. Since concept nodes may introduce further sub-dimensions, the CDT is not limited to a simple taxonomy but provides a representation in which contextual information can be progressively refined according to the needs of the application [27].

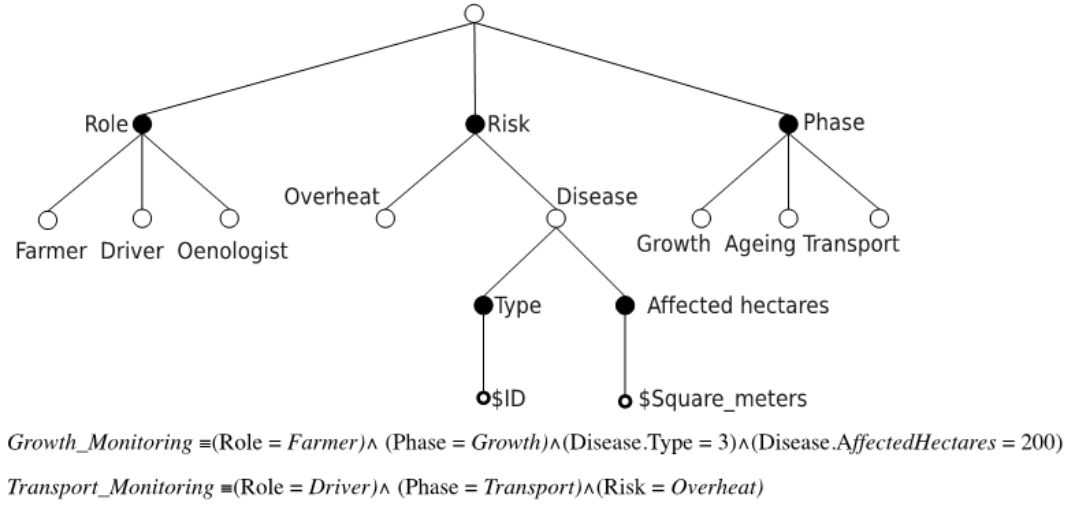


Figure 3.4: Example of Context Tree from the PerLa system [28], with two possible contexts from this tree.

Within this framework, a *context* is defined as a conjunction of *context elements*, where each context element is expressed in the form *Dimension = Value*. A specific context therefore corresponds to a particular subtree of the CDT, obtained by selecting one concept for each relevant dimension while preserving the hierarchical relationships encoded in the model. Sibling concepts belonging to the same dimension are mutually exclusive, whereas different dimensions cooperate to provide a complete description of the situation under analysis.

One of the main advantages of the CDT lies in the explicit separation between symbolic and numerical information. Numerical observations, typically obtained from sensors or external data sources, can be translated into symbolic concepts through suitable interpretation rules; for example, a measured temperature may be mapped to concepts such as *cold*, *mild*, or *warm*. This allows contextual reasoning to operate at a semantic level independently of the original numerical values, improving both human interpretability and context-aware decision-making.

The CDT has been successfully employed in several context-aware systems because it provides a flexible mechanism for modelling heterogeneous environments while maintaining a clear conceptual structure. In the context of this thesis, it represents the conceptual foundation for modelling the different contexts associated with an urban environment and for connecting environmental conditions, detected urban elements, and higher-level analytical representations within a unified framework.

4 | Automatic Computation of Aesthetic Indicators

This chapter illustrates how the B.7 and B.11 KPIs of the New European Bauhaus can be formalised and implemented in Python starting from an unstructured image dataset. The input dataset consists of street-level photographs of building facades, which are processed through a Computer Vision pipeline to automatically compute the numerical values associated with the two indicators, according to the definitions provided in the NEB self-assessment tool. The formalisation of the indicators is inherited from the previous thesis; the contribution of this work lies in the automation of the entire computational pipeline, enabling the automatic extraction of the required information from raw images. The following sections describe each stage of the proposed pipeline, from dataset preparation and facade extraction to the computation of the final indicator values.

4.1. Dataset Preparation

The starting point for the implementation of this part of the project was a dataset of buildings' photographs which were taken from the streets, but the bare images themselves cannot be considered ready to be analysed for the calculation of parameters inside the functions, since they present many issues. In primis, photos taken from the street return a modified prospect of the building's facade and need to be warped in order to return a perpendicular visual of the surface. Other adjustments can be applied to the picture (for example, colour saturation or shadow detection), but for the aim of the thesis, these preparations are irrelevant.

4.1.1. The dataset

As in [18], the dataset is composed of photos taken from the Lambrate neighbourhood in Milan Figure 4.1. The reason why this dataset was kept is that all the buildings' facades are coherent with each other, so it is simpler to assess coherency indicators.



Figure 4.1: Examples of images from the dataset.

Moreover, one important aspect of the set of images comes from the extension of the files, which is HEIF: this particular iPhone extension bundles a series of metadata that can be useful to address many concerns related to indicators' computation. For example, it contains important information about the position where the photo was taken, or the direction of the phone when the photo was taken. Other information concerns, for example, the light or the saturation of the image, which can be useful to correct chromatic aspects, leading to a more precise computation of aesthetic indicators. The transition from HEIF to JPG file might cause the loss of all this important metadata, and this is the reason why during the implementation of this project it was necessary to be sure not to lose all the important metadata while sharing the images with a different device.

4.1.2. Automation of the Process

Once the dataset had been collected in one of the project's folders, the focus went on the cleaning and normalisation of the numerous images. The photos, which were taken from the streets, had a perspective from below, which made the facade appear to be trapezoidal-shaped. In order to have a clear perspective on the elements that compose the building, it was important to rectify the perspective of the facade to conserve the physical parallelisation of the lines that composed the facade and to generate a frontal elevation that was suitable for the analysis. In the previous project work, this process was initially done manually, and a semi-automatic pipeline was later implemented, which would take the original image as input and, after a manual rectification and cropping, would return the processed image to the user. The aim of this project instead is to automate the whole process, so that the system would return a normalised image, which

would later be processed and analysed to compute the various indicators. In order to do so, the implementation must be able to both recognise the shape of the building by itself and to normalise it inside a rectangle, which would not alter the relative dimensions of the facade.

4.1.3. SAM3 for Facade Rectification

In order to automatically identify the shape of the building facade, a segmentation model was required. Traditional object detection models are able to recognise the presence of an object and return a bounding box around it, but this level of precision is not sufficient for the purpose of rectification. In fact, to correctly normalise the prospect of the building, it is necessary to know the exact contour of the facade and not only its approximate position inside the image. For this reason, the Segment Anything Model 3 (SAM3) [24] was adopted. SAM3 is a foundation model developed for image segmentation tasks, and it is able to generate masks for objects contained in an image. Unlike traditional segmentation models, which are usually trained to recognise a limited number of predefined classes, SAM3 is designed to segment generic objects and can therefore be applied to a wide variety of scenarios without requiring additional training. Given an input image, the model analyses the visual content and generates one or more masks representing the objects detected in the scene. Each mask corresponds to a set of pixels which are considered to belong to the same object; the result is therefore a much more precise representation of the object compared to a simple bounding box. Once the aim of the implemented process is explained, this section describes the main steps of the pipeline for the rectangularisation of the input image.

1. The automatic rectangularisation process starts by preparing SAM3 files to detect the building inside the original image.
2. The model generates one or more segmentation masks, among which the largest one is selected, assuming that it corresponds to the main facade.
3. The mask is then refined by filling possible holes and by applying morphological operations, so that a cleaner and more continuous facade region can be obtained.
4. Once the mask has been extracted, its external contour is analysed in order to estimate the four main corners of the facade. Instead of simply using the bounding box of the detected object, the algorithm fits lines along the left, right and upper borders of the mask: The intersections between these lines are used to identify the upper vertices, while the lower vertices are estimated using the lower limit of the contour. In this way, the trapezoidal shape caused by the perspective distortion can

be approximated more accurately.

5. The four detected points are then ordered and used as the source points for a perspective transformation. A destination rectangle is created by computing the maximum width and height of the detected facade, so that the resulting image preserves the relative proportions of the original building.
6. Finally, a homographic transformation is applied to warp the original image into the rectangular destination plane.

The output of this process is a rectified image, together with the corresponding rectified mask, which can be used in the following stages of the pipeline. A detailed pipeline framework is described in Figure 4.2.

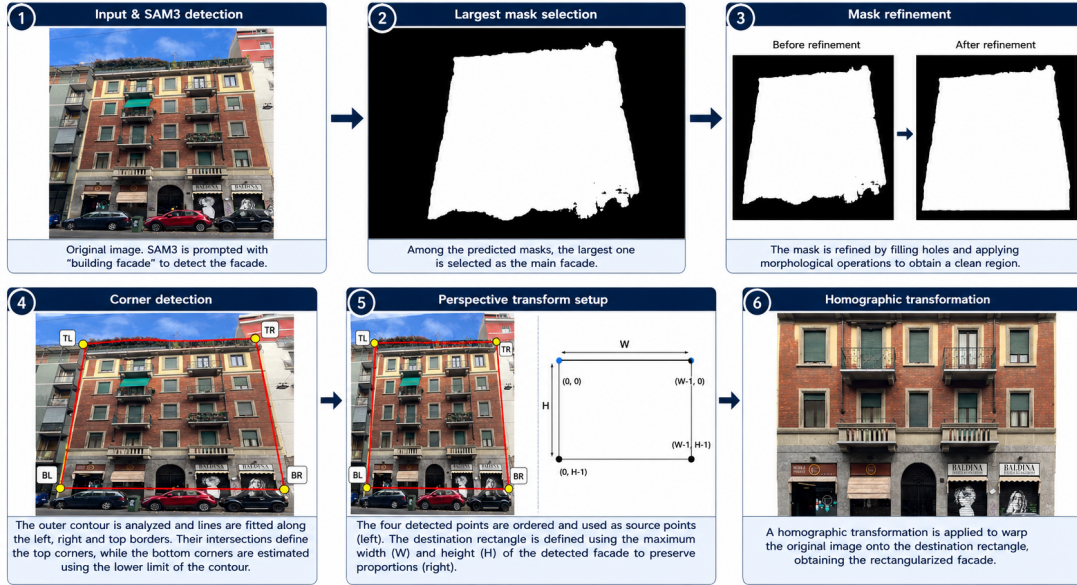


Figure 4.2: Facade rectangularisation pipeline.

4.1.4. Manual vs Automated Dataset Preparation

After developing this preliminary part of the implementation, a benchmark between the semi-automatic and the automatic approach needs to be assessed. The automation of the process brought many upgrades to the overall pipeline of the analysis of the picture, preventing human biases from impacting the process. Moreover, it standardises the edge selection of the building facade, leaving no interpretation to the user to what a facade is or which elements to be considered for the analysis. On the other hand, some downsides are clear for this kind of approach: for instance, there is an inefficiency based on the selection of the main facade, since the warped image can comprehend small sections of

other buildings, which can heavily impact the subsequent steps (this issue will be better addressed later). In terms of time inefficiencies, the process showed no relevant issues, since in the semi-automatic approach, a good amount of time was used to warp and crop the image, while on the automatic approach, a similar amount of time, around half a minute, was used to find the main facade, while the time to rectify the image can be considered as irrelevant. Overall, the precision of the two approaches can be considered comparable: maybe the human eye is more trained to recognise the edges of a facade, but as it was mentioned before, the semi-automatic method may give the user some space for interpretation and errors in pointing out the edges of a building can be made. Once the techniques behind the rectangularisation of the buildings' facades are understood, it is time to delve into the main section of the chapter, which is also the most relevant in terms of change of technique, compared to the previous work.

4.2. Architectural Elements Recognition

At the beginning of the original project, all the NEB indicators in the Beauty section were taken into account but, as a result of a thorough analysis, only some of them were considered computable with the data coming from the image of a facade. In order to numerically estimate these, it was vital to extract the individual elements that composed the facade and to categorise them for further analysis. This passage was initially carried out manually, leveraging a programme called Roboflow [29], where it is possible to manually draw rectangular bounding boxes around the elements; once the categorised bounding boxes were selected, they were exported in a JSON format to Python, to later be used with the integration of a parser .py function. This kind of application was time-consuming from the programmer's side, and it also prevented the overall system from being scalable: this is the reason why, in this thesis, the process has been automated.

4.2.1. Automatic Recognition

The pipeline for the recognition of architectural elements inside an image is carried out initially with the SAM3 library. The elements to return as a list of bounding boxes were the same as the previous thesis: in particular, windows, balconies, shops, entrances, French doors, and graffiti (or street-art in general). At the beginning, the code was structured to have a single function to recognise each type of element of the facade, but, even if this approach might be effective from an operational point of view, it soon turned out to be very inefficient due to long waiting times and relevant computational overhead. This problematic behaviour of the code was a result of the multiple segmentations that

were initialised on the same image, which would cause an enormous delay for each run of the code. Moreover, this approach turned out to be critically inefficient every time it was necessary to debug the code. To solve this relevant issue, all the single functions were merged into one, so that the pre-processing of the image would be carried out only once for all the elements to find, leading to a more efficient use of the resources and to a more scalable solution. Even if this strategy would save much time and resources, there was still a bottleneck in the process of implementing the whole function: when it came to debugging, the time to wait was still too long (around 1 and a half minutes for each debug) and the preparation of variables coming from the SAM3 library took much time.

To overcome the previous issue a new strategy was adopted: we implemented the exploitation of the cache to avoid calculating the analysis of the same data for multiple times. Basically, this technique consisted in using a Boolean parameter, which was passed to the main function *'get building elements'*: if the parameter *'cache'* was False, then the operating system would recompute all the bounding boxes for the image path given to the function and then save the results in the cache at the end of the process, whereas, if it was True, the operating system would skip the SAM3 image processing and retrieve the bounding boxes from the cache. It is important to underline that this Boolean parameter is completely independent from the image path given as input, so it is important to set it correctly; it might return bounding boxes belonging to different images from previous runs on completely different images. So, if it is not a matter of debugging, the parameter must always be set as False.

Another issue with this kind of technique was related to one of the parameters given to the SAM3 semantic predictor: the confidence of the model. In fact, from the various debugging sessions, it became clear that, based on the confidence set in the code, the program would either ignore many trivial elements (high confidence), for example, many windows with closed shutters were not recognised, or it would return wrong and redundant bounding boxes. It is important to take into consideration that for this part of the thesis, it is vital to be quite precise with element recognition, otherwise indicators like symmetry would be completely biased. This is the reason why the decision fell on keeping a low confidence value of 0.3 out of 1, with which every object is usually detected, but the bounding boxes are redundant, so one single object can have multiple bounding boxes, and some of them do not actually match their type, for example, a car window can be recognised as a building window. It is clear at this point that if there is the necessity to have certain standards of precision, that first result cannot be considered as sufficient, and that a more refined way to categorise the bounding boxes must be implemented. SAM3 is a powerful tool to interpret the semantics of a single word, but it is not precise when it

comes to integrating meaning and syntactic recognition of a whole sentence, so the bias cannot be fixed by describing the architectural elements in a more specific way. The best way to overcome the redundancies and the misinterpretations was to implement functions that could force the final set of bounding boxes to be as accurate as possible.

- For instance, we implemented a function which can detect redundancies and recognise multiple boxes insisting on the same element: the function groups the overlap rectangles and keeps only the biggest one, so that the user will get the elements containing all the parts. For example, one of the most redundant outputs was related to windows: SAM3 used to select both a bounding box for the bare glass part of the window and one for the whole element, cornice included. The *'check for duplicates'* function would keep only the last one.
- Another important function was the *'check no car windows'* one, which, as one can realise from the name, checks the presence of windows belonging to vehicles and it deletes them from the list of bounding boxes. To implement this function, it was necessary to use the semantic interpreter provided by SAM3 to find out cars and vans on the image.
- The same logic as the previous function was applied to the *'no windows on shops or entrances'* function, which opts out all the windows belonging to the ground floor display windows type, so basically all glass elements from shops and entrances.
- One more important issue needed to be solved: the semantic predictor struggled to distinguish 'normal' windows from French windows (glass doors on balconies), and it also failed to recognize the real height of the window-door, thus another function was implemented. It recognizes a French window based on the relative position between a window and a balcony, adding a mandatory tolerance so that no false negative is left out of the list. The bounding boxes' height is also adjusted so that the element is approximated to its real dimension.
- Finally, a function was implemented to try and distinguish shops from building entrances when the semantic predictor recognizes an element as belonging to both types. The algorithm aims at recognising which is the main entrance, starting from the assumption that usually there is just one single entrance to each building, while the other ground floor openings are usually commercial activities.

In the Figure 4.3 we can clearly see the difference, in terms of precision, between windows detection with only SAM3 and adjusted windows detection with proper tailored functions.

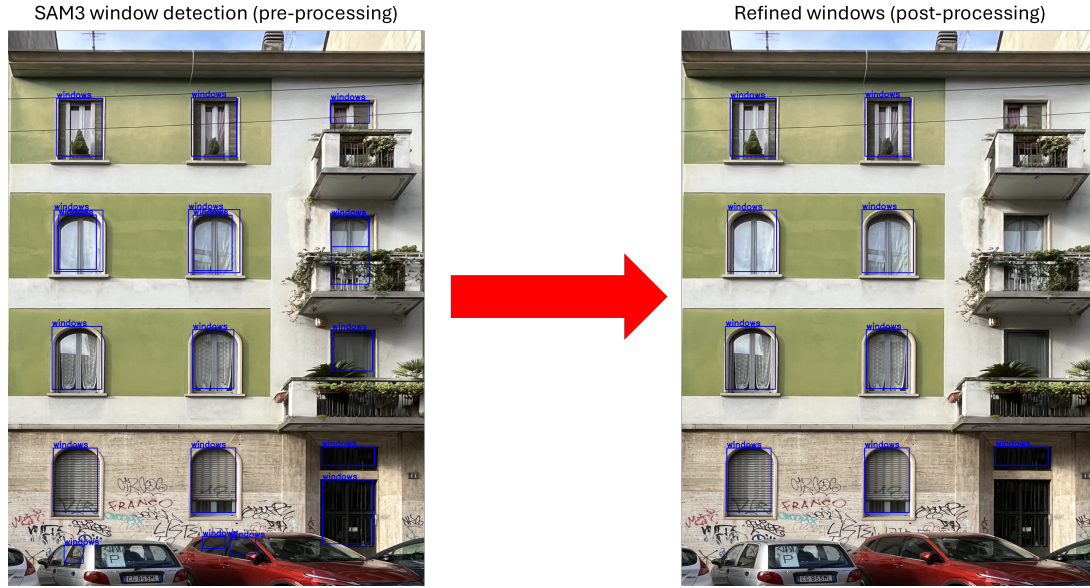


Figure 4.3: Architectural element recognition refined.

To summarize, it is important to stress the fact that this automatic recognition pipeline of architectural elements of a building does not stop on the raw bounding boxes provided by the semantic predictor, but tries to process them in order to refine the precision of the final output. This hybrid approach allows the system to exploit the flexibility of a foundation model while maintaining a sufficient level of precision for the computation of the aesthetic indicators presented in the following chapters.

4.2.2. Manual vs Automated Architectural Elements Recognition

Once the main operations of the automated pipeline are highlighted, it is important to benchmark this approach with the one exploited on the previous project. In the original thesis, the bounding boxes were retrieved manually, drawing the rectangular bounding boxes around the various types of elements with the Roboflow program, to be later interpreted by a proper parser designed in Python. This former method is surely more accurate, since it is human supervised, so it is even possible to annotate elements which are hidden in the photo which a human eye can recognise, but on the other hand, it presents many limits to the overall pipeline of the project. For instance, the human effort to draw the bounding boxes is very time-consuming, and this process prevents the whole implementation from being scalable, since each image must be pre-processed every time before delivering any tangible results. Moreover, to implement KPIs about architectural coherence in an urban context, it is vital to have a large dataset, which can only be

processed with an automated tool to be timely and efficient.

4.2.3. Pipeline Efficiency Issues

Theoretically, at this point of the project, the pipeline would appear to be, even with some minor drawbacks related to both the rectangularisation of the and the recognition of the elements, fully automated and perfectly scalable, but unfortunately, it presents a grave issue, which is not easily solvable: this is about time efficiency. This first step of the pipeline takes, on average, one and a half minutes for each building to be completed, and if there is the need to study the indicators of a whole neighbourhood, this is not sustainable. The bottleneck is due to the semantic prediction of SAM3, which, as I mentioned before, despite providing a satisfactory level of accuracy and flexibility, requires a significant number of computational resources. As a consequence, processing hundreds or even thousands of facades would result in excessively long execution times, limiting the practical applicability of the proposed approach. For this reason, a second phase of the research was devoted to investigating whether the architectural elements automatically detected by SAM3 could be exploited to generate a labelled dataset suitable for training a dedicated object detection model. The underlying idea was to use SAM3 as an automatic annotation tool and subsequently train a YOLO model capable of recognising the same architectural elements with a significantly lower computational cost. If successful, this strategy would preserve most of the automation introduced in this thesis while drastically reducing inference times, making the overall pipeline more suitable for large-scale urban analyses.

4.3. YOLO for Cross-Validation

In the previous thesis, there has been an attempt to find the architectural elements on the building through a particular time-efficient Python library, YOLO. The results were not satisfactory, since the algorithm could not detect the bounding boxes efficiently, and that was the reason why that method was put aside. On the other hand, YOLO [25] is not just a library that, given the name of the element to find inside an image, is able to return the corresponding bounding boxes, but it is also a powerful tool to train a recognition model through machine learning. YOLO, which stands for You Only Look Once, is one of the most widely adopted object detection frameworks in computer vision. Its main advantage lies in the ability to perform object localization and classification simultaneously through a single neural network; this makes the model particularly suitable for applications where computational efficiency is required. Furthermore, YOLO can

be trained on custom datasets, allowing the recognition of specific categories of objects that are not included in the original training set. Different versions of the framework have been developed over the years, progressively improving detection accuracy, inference speed, and robustness to variations in scale, illumination, and perspective. Because of these characteristics, YOLO has found applications in several domains, ranging from autonomous driving and surveillance systems to urban analysis and remote sensing. In the context of this thesis, its capability to learn custom visual features makes it a suitable candidate for the recognition of architectural elements from images.

Due to the limited size of the proposed dataset, which contains only 99 images, the cross-validation technique was performed in order to train a model that can predict the output with sufficient confidence. While cross-validation defines the strategy adopted to maximize the use of the limited dataset available, the overall training process involves several additional stages, ranging from the generation of the annotations to the final evaluation of the detector. Before training the model, the images and the corresponding bounding boxes obtained through the SAM3-based pipeline must be converted into the format required by YOLO and organized according to its dataset structure. The complete pipeline adopted in this thesis, including data preparation, cross-validation, model training, and performance evaluation, is summarized in Figure 4.4.

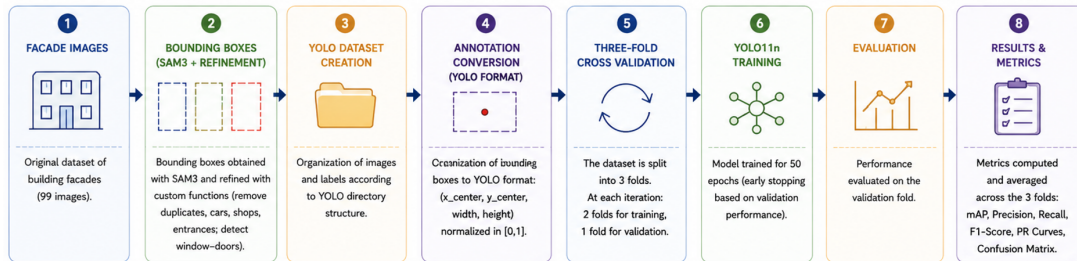


Figure 4.4: YOLO cross-validation pipeline.

4.3.1. Data Preparation

Before training the model, it was necessary to prepare the dataset according to the requirements of the YOLO framework. The available annotations had originally been produced during the previous stages of the thesis and were therefore not directly compatible with the format expected by the model. For this reason, a dedicated preprocessing function was implemented. The first step consisted of associating each architectural element with a numerical class identifier. Windows, entrances, balconies, shops, and graffiti were selected as target categories, while French windows were intentionally excluded from the training process and merged with the windows category during the annotation stage. The result-

ing class structure was then used to generate the labels required by YOLO. Afterwards, the coordinates of each bounding box were converted into the normalized representation adopted by the framework. Instead of storing the coordinates of the corners of the rectangle, YOLO requires the coordinates of the center point together with the normalized width and height of the object. Finally, for each image, a corresponding text file containing all the annotations was generated automatically. The images and labels were then organized into the folder structure expected by YOLO, producing a dataset that could be directly used for training. Although conceptually simple, this preparation phase required considerable attention, since even small inconsistencies in the annotations could compromise the entire training process.

4.3.2. Implementation

A specific Python function was implemented to train the model on the dataset. At first, it was important to test the function itself to understand if the machine learning process would actually work, so the number of epochs, which is the number of times the model iterates through the entire training dataset, was intentionally kept low. In particular, we used only 10 epochs during the first experiments. Even with this limited configuration, the training process required approximately one and a half hours to complete. This was mainly due to the fact that the computation was executed on a personal computer without a dedicated GPU; therefore, all the operations were performed on the CPU, significantly increasing the training time. The implementation consisted of two main phases: *dataset preparation* and *model training*:

1. First, the original dataset was reorganized according to the requirements of YOLO. For each image, the corresponding annotation file containing the bounding boxes of the architectural elements was generated and stored using the YOLO format. The elements considered for the training process were windows, entrances, balconies, shops and graffiti (n.b.: French windows were left out, since they were forced manually with the code on the windows bounding box list). Afterwards, the cross-validation procedure automatically divided the dataset into three folds, creating a different training and validation set for each iteration. In this way, every image was used both for training and validation during different stages of the process.
2. Once the dataset had been prepared, the training phase could start. A pre-trained YOLO11n model was selected as the starting point; this choice was motivated by its relatively small size and reduced computational requirements, which made it suitable for the available hardware resources. For each fold, the model was trained

independently using the corresponding training set and then evaluated on the validation set. The image resolution was fixed at 416 pixels and a batch size of 4 images was adopted in order to limit memory consumption. Furthermore, an early stopping mechanism was introduced so that the training could be interrupted automatically if no improvements were observed for a certain number of epochs.

The objective of this first experiment was not to obtain the final model, but rather to verify whether the available dataset contained enough information for the network to learn meaningful visual patterns. If satisfactory results had been obtained, the same procedure could then be repeated with a larger number of epochs and more computational resources, allowing the model to further refine its capability to recognize architectural elements on building facades.

For the second experiment, we used another hardware, with a dedicated GPU, to train the model, and the number of epochs was set to 50 instead of 10. The terminating algorithm was kept in order to stop the execution if no improvement was detected during the training of the model. This time, with the new hardware, the process took far less than the first time: in around 45 minutes the overall experiment was done, highlighting the benefits of a GPU when it comes to a machine learning algorithm.

4.3.3. Results

The performance of the detector was evaluated through a three-fold cross-validation procedure. Two different training configurations were tested: an initial experiment consisting of 10 epochs, aimed at verifying the feasibility of the training process, and a second configuration with 50 epochs, intended to assess the actual capabilities of the model once a sufficient convergence had been reached.

The preliminary experiments immediately highlighted the need for a longer training phase. After only 10 epochs, the detector was capable of recognising mainly windows and balconies, while entrances and shops were almost never detected correctly. This behaviour can be observed both in the confusion matrices and in the Precision-Recall curves (Figure 4.5), where the Average Precision associated with these two classes remains close to zero. Consequently, although the model was already capable of identifying the most visually distinctive facade elements, its overall performance was not sufficient for a reliable deployment.

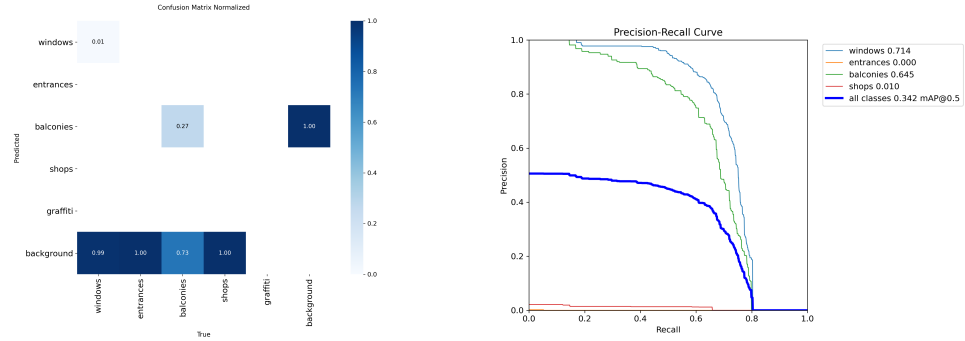


Figure 4.5: Normalised confusion matrix and precision recall matrix with 10 epochs.

A significant improvement was observed after increasing the number of epochs. In all folds, windows and balconies achieved the highest performance, with Average Precision (Figure 4.6) values consistently exceeding those obtained in the preliminary experiments. In particular, windows reached values close to 0.85 in the best-performing fold, while balconies achieved values close to 0.80. Entrances also showed a substantial improvement, reaching Average Precision values of approximately 0.40, whereas shops remained the most challenging category despite a noticeable increase compared to the initial configuration. The confusion matrices further confirm these observations. Most of the correctly classified instances belong to windows and balconies, while a considerable number of entrances and shops are still associated with the background class.

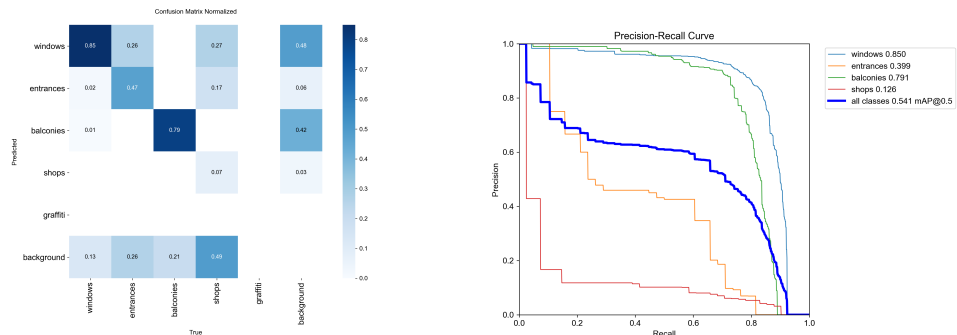


Figure 4.6: Normalised confusion matrix and precision recall matrix with 50 epochs.

This behaviour is not entirely unexpected. On the one hand, windows and balconies are among the most recurrent architectural elements in the dataset and usually exhibit relatively stable visual characteristics. On the other hand, entrances and commercial activities present a much higher variability in terms of dimensions, appearance and visibility within the image, making their identification intrinsically more difficult. Another

should be drawn manually to provide a precise dataset to the YOLO model: this way, since the model works well with the cross validation techniques, as it was demonstrated through the project, it can be trained to recognise architectural elements with a higher accuracy.

4.4. Visual Richness Indicator

The Visual Richness indicator (B.7) is composed of 4 different parameters, which are related to the shape of the facade, the symmetry of the facade, the composition of the facade's architectural elements, the presence of shops, and the presence of street art or graffiti.

The latter aspects have been computed simply leveraging SAM3, and in the case of the shops, they also have been processed according to the techniques described in the previous chapter: the two indicators are equal to one if the system detected their presence, otherwise they are equal to zero.

The parameter O, related to the overall order of the facade, is the component that underwent the most substantial modifications with respect to the previous implementation. In the original version of the project, the symmetry analysis was performed through a collection of dedicated functions; each architectural element was analysed independently, and different procedures were required to evaluate windows, balconies, window-doors, and ground-floor elements. The underlying principle was based on the identification of mirrored bounding boxes with respect to a symmetry axis computed directly from the dataset annotations. Although this approach produced satisfactory results, it presented a number of limitations: the code was highly dependent on the structure of the dataset, and several portions of the logic were repeated across different functions. Furthermore, every new architectural element required the development of an additional dedicated procedure, reducing the overall scalability of the framework. For this reason, the symmetry assessment was redesigned in the current implementation. Rather than searching for corresponding pairs of bounding boxes individually, the detected elements are first organised according to their spatial distribution on the facade; rows and columns are generated and subsequently used to create a simplified representation of the architectural composition. The analysis is therefore no longer focused on the individual object itself but on its position within the overall structure. This allows the system to evaluate symmetry through a more general framework which can be applied to different configurations without requiring element-specific adaptations. At the same time, the information generated during this process can also be reused by other indicators, reducing computational redundancy and

improving the integration between different stages of the pipeline. A second contribution to the parameter O is related to the geometrical regularity of the facade. In the previous version of the project, this aspect was evaluated directly on the segmented prospect by comparing the area of the contour with the area of its enclosing rectangle; the closer the facade was to an ideal rectangular shape, the higher the resulting score. While effective, this method was inevitably influenced by perspective distortions and by the quality of the original image. Within the current framework, this limitation is partially addressed through the rectangularisation process described in the previous chapter. Since the facade is first normalised and projected onto a frontal plane, the subsequent geometrical evaluation is performed on a representation that is considerably more coherent with the actual shape of the building. As a consequence, the indicator becomes less sensitive to camera position and viewing angle; an aspect which is particularly relevant when dealing with large collections of images acquired under heterogeneous conditions. In Figure 4.8 the symmetry implementation pipeline is represented.

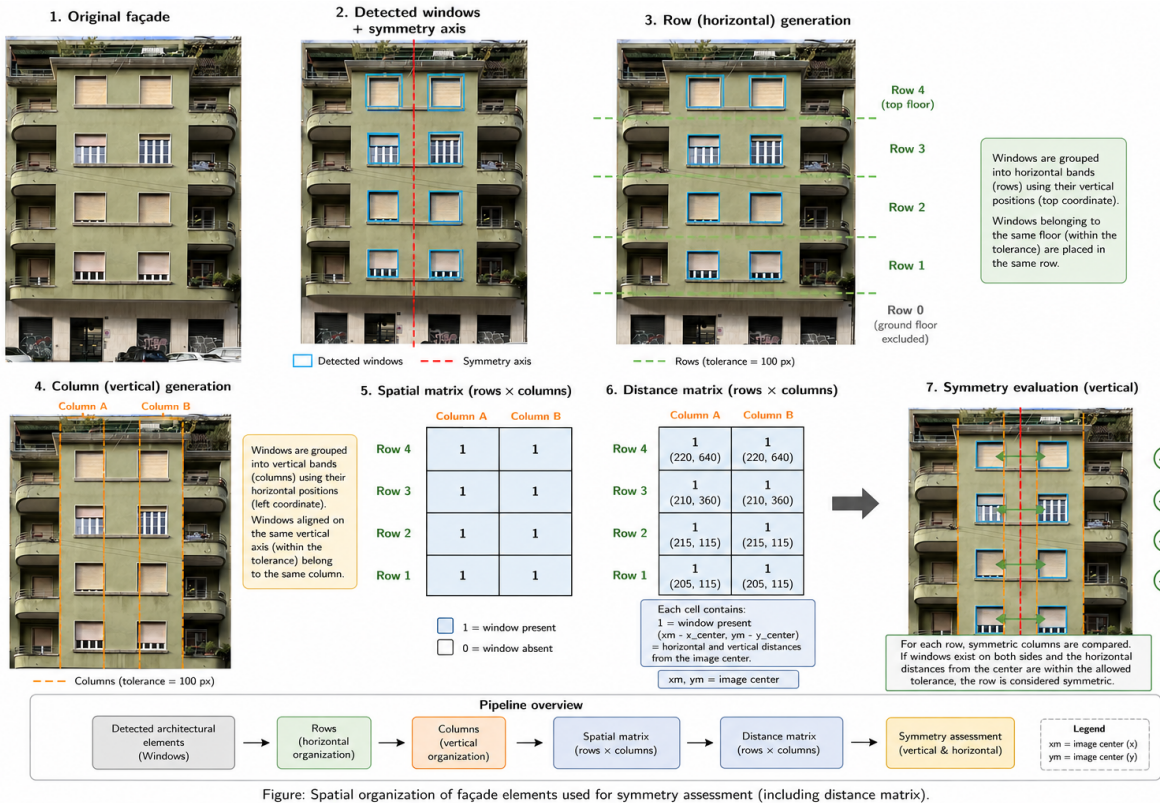


Figure: Spatial organization of façade elements used for symmetry assessment (including distance matrix).

Figure 4.8: Example of the symmetry implementation pipeline.

The parameter C, which describes the compositional characteristics of the facade, also required several modifications. One component of this score is still represented by the ratio between openings and opaque surfaces; this metric proved sufficiently robust in

the previous implementation and therefore its underlying logic remained substantially unchanged. The total area occupied by windows and window-doors is compared with the overall area in order to estimate the degree of visual permeability of the building envelope. This information is particularly useful because it provides a simple quantitative description of the composition while requiring only the bounding boxes generated by the detection pipeline.

More significant changes were instead introduced in the evaluation of the architectural rhythm of the facade. In the previous implementation this contribution was approximated through a relatively simple condition: the indicator only verified the presence of a sufficient number of balconies and used this information as a proxy for the existence of horizontal or vertical compositional patterns. Although computationally inexpensive, this solution provided only a partial representation of the actual organisation of the and was unable to capture more complex arrangements of architectural elements. The current version adopts a broader perspective: rather than focusing on the presence of a single category of objects, it analyses the overall distribution of openings and recurring architectural components. By exploiting the same spatial organisation generated for the symmetry assessment, the framework is able to identify regular alignments and recurring compositional schemes that contribute to the perception of order within the facade. The resulting indicator therefore captures a wider range of architectural configurations and provides a more representative description of the compositional structure of the building.

It is also worth mentioning that not all the components of the original framework required substantial revisions. The colour analysis employed within the B.11 indicator remained largely unchanged; unlike the other functions, it already operated directly on image information and therefore did not depend on the annotation structure adopted by the previous dataset. For this reason, the dominant colour extraction procedure and the associated colour-distance computation could be integrated into the new pipeline without significant modifications. This aspect considerably simplified the migration process and ensured consistency between the results obtained in the two versions of the framework.

4.5. The Aesthetic Perception Indicator

The implementation of the Aesthetic Perception indicator (B.11) indicator required considerably fewer modifications compared to B.7. This KPI is based on two main aspects: the geometrical regularity of the facade and its chromatic composition. Since both contributions can be extracted directly from image information, the transition to the new computer vision pipeline was relatively straightforward. Similarly to the B.7 indicator,

the geometrical component now benefits from the rectangularisation process introduced in the current framework. By evaluating the shape of a normalised prospect rather than the original image, the influence of perspective distortions is reduced and the resulting score becomes more robust across different acquisition conditions. The colour analysis, on the other hand, remained substantially unchanged. The original implementation was already fully compatible with the new workflow, as it directly operates on image pixels rather than on manually annotated bounding boxes. The algorithm identifies the dominant colours of the facade and evaluates their relative prevalence within the image; this information is then used to estimate the visual coherence of the building from a chromatic perspective. Consequently, the main evolution of the B.11 indicator concerns the integration of the normalisation stage, while the colour extraction procedure was preserved from the previous version of the framework due to its effectiveness and generality.

5 | The I.3.7 Indicator: Accessibility to Public Transportation

The third section of the New European Bauhaus is dedicated to the definition of inclusivity in urban contexts and comprehends nine different indicators to evaluate it. Among them, Indicator I.3.7 focuses on the accessibility to public transportation, with the objective of assessing how easily citizens can reach and use the available transport network[9]. Accessibility plays a fundamental role in the concept of urban inclusivity, since it directly influences the possibility for people to access services, workplaces, educational opportunities and social activities regardless of their location within the city. Within this thesis, the computation of I.3.7 represents a substantial change with respect to the KPIs presented so far. While the input dataset is still composed of images of building facades, the focus is no longer on their visual representation, but rather on the information that can be derived from their metadata. The aim is to evaluate the accessibility of the building to the public transportation network and to express this assessment through a single score ranging from 0 to 100.

The inspiration for computing this indicator came from the NEB document, where specific thresholds (Figure 5.1) are proposed to evaluate the quality of accessibility to public transportation; however, since the study case of this thesis is Milan, it became clear that these thresholds needed to be redesigned, as they proved excessively generous for a dense metropolitan environment. More generally, such limits can be considered too rigid, since they do not take into account the context of the urban analysis; it is evident that a large metropolitan area cannot be evaluated through the same parameters adopted for smaller urban or rural contexts. Consequently, the original NEB framework was taken as a starting point rather than being adopted directly. The objective was to preserve the rationale behind the indicator while adapting its computation to the characteristics of the study area and to the information available within the proposed pipeline. The resulting methodology aims to provide an accessibility score that is coherent with the

urban structure of Milan and that can be automatically computed for every building included in the dataset.

Metric: What is the level of pedestrian accessibility ¹ to public transport? (single selection allowed)				
1. The average distance to public transport stops is more than 800 m.				
2. The average distance to public transport stops is between 500 and 800 m.				
3. The average distance to transport stops is between 300 and 500 m.				
4. The average distance to transport stops is less than 300 m.				
Indicator score	[1]	[2]	[3]	[4]
	0	50	70	100

¹ Pedestrian accessibility to public transport evaluated according to Section 5.6.8.

Figure 5.1: Original KPI thresholds from the NEB.

5.1. Data Acquisition

As mentioned before, the computation of the accessibility indicator changes dramatically compared to the previous ones, since the image alone can no longer be considered sufficient. While the KPIs discussed so far relied almost entirely on information directly observable in the facade, the evaluation of transportation accessibility requires knowledge that extends beyond the visual content of the photograph itself. First of all, it is necessary to extract information related to the context in which the photograph was taken; secondly, this information must be integrated with another source of knowledge capable of describing how that specific building is contextualised within its surrounding environment. In other words, metadata coming from the image must be extracted and analysed together with structured information obtained from relational datasets. This process represents a fundamental step of the methodology, since it enables the transition from the analysis of an isolated building to the analysis of the urban system in which it is embedded. By combining multimedia and structured data, it becomes possible to associate each building with information regarding the transportation infrastructure available in its surroundings and, consequently, to evaluate its level of accessibility. For this reason, before presenting the actual computation of the KPI, it is necessary to describe how contextual information is extracted and how heterogeneous data sources are integrated into a common framework.

5.1.1. Metadata from Images

The first step of the process consists in extracting the metadata associated with the image. This operation introduces an important requirement concerning the type of files that can be processed by the framework: the photographs must preserve their EXIF (Exchangeable Image File Format) information. Modern smartphones and digital cameras automatically generate these metadata during image acquisition and store additional information along-

side the image itself EXIF metadata can contain a wide variety of information, ranging from camera parameters and acquisition time to image orientation and geographical position; among these attributes, the most relevant for the purpose of this thesis are the GPS coordinates associated with the image. When location services are enabled on the device, latitude and longitude are embedded directly within the file and can therefore be retrieved automatically. The implemented procedure analyses the EXIF structure of the image and extracts the GPS tags containing the geographical coordinates. The availability of this information is fundamental because it allows the image to be positioned within the urban environment and, once the geographical coordinates have been retrieved, they can be used to access additional relational datasets describing the surrounding context, thus enabling the integration between multimedia and structured information sources.

5.1.2. Structured Datasets

Data was acquired from the official web portal of the Comune di Milano, called Dati.Milano [30], which provides the user with a list of many datasets in various formats, such as .csv or .json. In this list, there was much information that can support projects related to urban design: in this case, collections of datasets about the underground transportation and the surface transportation were particularly of interest. The choice fell on two specific datasets: one of which collected all the relevant information about the metro stops [31] and the other contained the list of trams and buses' stops [32]. The first dataset was in csv format, while the second one was in json format, so the integration with the rest of the parameters of the code was quite simple, since they were both perfectly structured datasets. The cleaning process was straightforward to do, since the tuples had all the necessary information to locate them on the urban space and no more useless data for the computation of the KPI. It was sufficient to rename the ids of the columns into a more comprehensible ones, for example from "LONG X 4326" to "longitude" and for the surface stops dataset to separate the traditional longitude and latitude format into two different columns.

It is needless to say that this kind of cleaning is specific to the type of dataset provided from the Comune di Milano website, thus the cleaning process was tailored to this specific structure. It is implied that at this point the overall pipeline designed for the computation of the I.3.7 KPI is not completely automated for every context analysed: this step of the process remains dataset-specific and it is not generalised. This is one of the main drawbacks of the thesis and one of the possible upgrades it might have in the future, to deliver a fully automated tool for every kind of different context.

5.2. Accessibility Analysis

Once both the geographical coordinates extracted from the image and the transportation datasets have been prepared, the accessibility indicator can be computed. The first operation consists in identifying the public transportation stops that are closest to the building represented in the photograph. To achieve this, the coordinates of the building and the coordinates of every transportation stop are converted into a metric reference system. This conversion is necessary because geographical coordinates expressed as latitude and longitude do not directly correspond to measurable distances; therefore, a transformation into meters allows the computation of Euclidean distances between points.

The process is performed separately for metro stations and for surface transportation stops. In both cases, every stop contained in the datasets is analysed and its distance from the building is calculated. Since a single transportation line may have multiple stops distributed across the city, only the closest stop for each line is kept. This choice avoids overestimating accessibility in situations where several nearby stops belong to the same transportation line and therefore do not provide additional connectivity opportunities.

After all distances have been evaluated, the stops are sorted according to their proximity to the building. From this ordered list, only a limited number of elements is selected: more specifically, the four closest surface transportation lines and the two closest metro lines are taken into consideration for the final assessment. This selection reflects the idea that accessibility should not be evaluated on the basis of every available stop in the surrounding urban environment, but rather on the transportation opportunities that can realistically be reached by a pedestrian, as it is highlighted by the inclusivity indicator from the NEB.

The final KPI is computed by assigning a score to each selected transportation line according to its distance from the analysed building. Distances are compared against a set of thresholds specifically designed for the urban context under examination. Depending on the distance from the stop, the line receives a score equal to 100, 70, 50 or 0: these scores refer to the New European Bauhaus standards. Buildings located close to several transportation lines, especially metro stations, obtain higher scores, whereas buildings characterised by poor connectivity to the public transportation system receive lower values. Moreover, different transportation modes do not contribute equally to the final indicator: surface transportation lines are assigned a lower weight, while metro lines receive a higher contribution. More specifically, each of the four selected surface transportation lines contributes 10% to the final score, while each of the two selected metro lines contributes 30%. As a consequence, the overall KPI can be interpreted as

a weighted combination of six transportation opportunities, where rapid transit systems have a stronger influence on the final evaluation than buses or trams. The KPI therefore offers a quantitative measure for urban accessibility that can be directly integrated into the broader framework of indicators developed throughout this thesis.

5.3. Parametrisation

Using the NEB’s specifics, the code would have assigned a score of 100 to a building situated at a distance lower than 300 meters from the closest station or stop, a score of 70 for a distance between 300 and 500 meters, one of 50 if it is between 500 and 800 meters, and 0 if it is more than 800 meters (Figure 5.2). Figure 5.3 shows the representation of all the buildings’ addresses in Milan, arranged by their coordinates. The different colours refer to the various levels of accessibility, where dark green represents an excellent accessibility position, light green represents good accessibility, yellow represents an acceptable accessibility and red represents low accessibility.

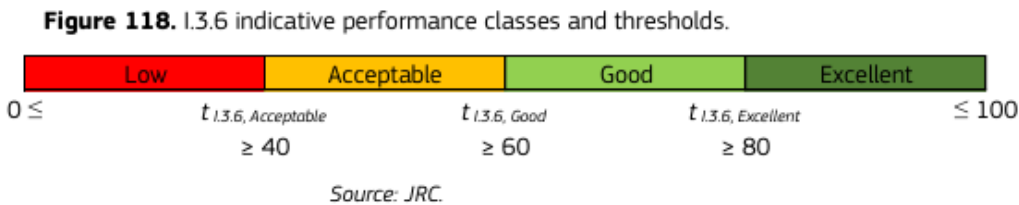


Figure 5.2: Original KPI classes and thresholds from the NEB.

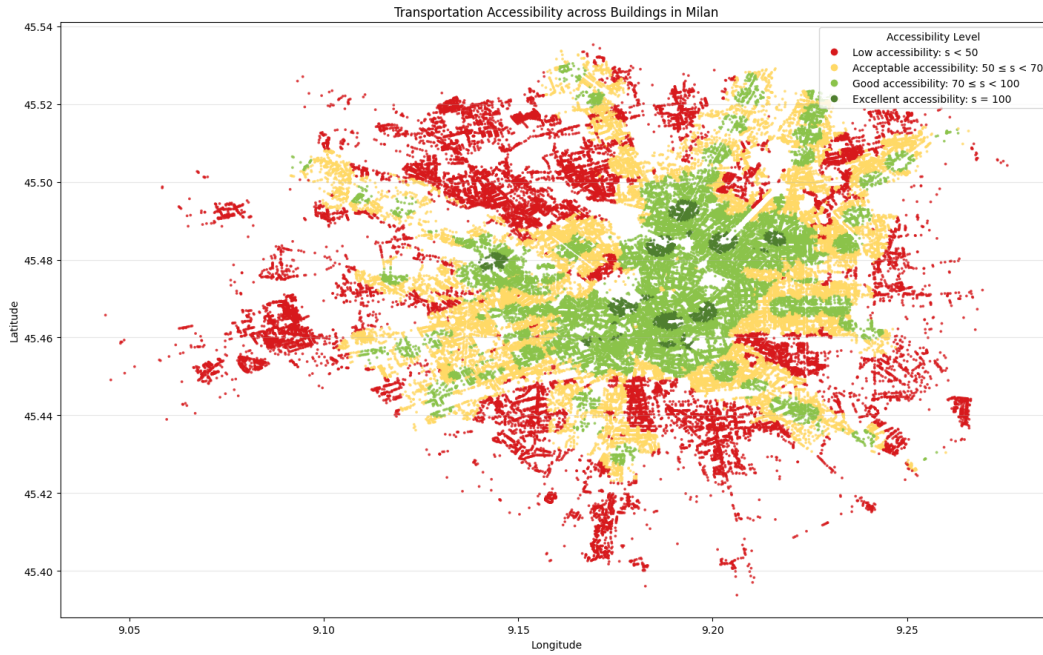


Figure 5.3: Representation of Milan’s buildings’ accessibility with the NEB original thresholds.

With these thresholds, due to the presence of a rich transportation network, all the buildings located in central Milan would have the same high overall score while those in the suburbs would have a lower score. Therefore, this kind of KPI might be a good indicator to differentiate urban contexts from rural contexts, but not to assess the real accessibility to public transportation from a building in a specific neighborhood. To overcome this discrepancy, the parameters need to be related to the area where the analysis is taking place, and this is the reason why a different adoption of the KPI computation has taken place. The solution we implemented starts from a simple consideration: if the transportation network is denser, then it should be more difficult for a building to obtain a high accessibility score. On the contrary, if the transportation network is sparse, it would be unfair to apply the same thresholds, since even relatively large distances might still represent a good accessibility condition for that specific context. For this reason, the first operation carried out by the function is to analyse the area surrounding the building. Starting from the coordinates extracted from the image metadata, the algorithm defines a square area extending one kilometre in every direction from the analysed point. This operation is not performed directly on latitude and longitude coordinates, since these do not correspond to real metric distances; instead, the coordinates are first converted into metres and the boundaries of the area are computed only afterwards. Once the limits have been found, they are converted back to geographical coordinates and used to interrogate

the transportation datasets. At this point, the algorithm counts how many metro stations and how many surface transportation stops are contained inside the selected area. The two datasets are analysed separately because, as explained before, they have a different impact on the final accessibility score. The result of this operation is not simply the total number of stops, but rather an estimation of their density inside the analysed portion of the city. The idea behind this choice is quite straightforward: if many stops are concentrated in a limited area, then accessibility should be evaluated with stricter criteria. On the other hand, if the transportation network is less dense, the limits should become more permissive. In this way, the KPI is no longer evaluated through fixed thresholds, but through parameters which depend on the urban environment surrounding the building. Once the density of metro stations and surface transportation stops has been computed, the algorithm derives a characteristic distance for each transportation category. This value can be interpreted as an approximation of the average spacing between transportation opportunities in that specific area of the city. The higher the density of stops, the lower this characteristic distance becomes; this way the thresholds used to evaluate the KPI become more demanding. Finally, three different limits are generated both for metro stations and for surface transportation stops. These thresholds replace the fixed values originally proposed by the NEB and are later used to assign the scores of 100, 70, 50 or 0 to each transportation line considered in the analysis. The final result is therefore an adaptive parametrisation of the accessibility indicator; instead of evaluating every building according to the same rigid rules, the KPI adjusts itself to the characteristics of the urban context where the analysis is taking place.

5.4. The Model

Taking into account the aspects highlighted in the previous section and related to the parametrisation of the thresholds, an overview of the model's limits and validation criteria is described in the subsequently two sub-sections of the thesis.

5.4.1. Limits

The solution described before, even if quite improved through the parametrisation of the thresholds, presents an important drawback in terms of efficiency. When it comes to selecting all the bus stops in the area, the calculator must take into account many stops in the dataset, and even if there is margin for improvement, it will never be enough to deliver a system that could compute the indicator for a long list of buildings. So, even if the model is precise, as will be shown in the next chapter, the scalability of it presents

some limitations in terms of time efficiency.

5.4.2. Validation

Before validating the model, a further optimisation was introduced to reduce the computational cost of the algorithm. In the original implementation, the distance between a building and every transportation stop contained in the datasets had to be computed repeatedly, resulting in excessively long execution times when analysing the entire city. To overcome this limitation, a spatial indexing structure based on the BallTree library was adopted. This allowed the algorithm to retrieve only the stops located within a pre-defined radius from the analysed building, significantly reducing the number of distance calculations while preserving the original logic of the indicator. Once the optimisation was introduced, the model was applied to all the buildings contained in the Milan dataset. The resulting accessibility scores were then represented geographically, assigning each building to one of four categories: low, acceptable, good or excellent accessibility. A qualitative validation of the results was performed through a comparison with a previous study conducted at Politecnico di Milano based on the PTAL (Public Transport Accessibility Level) methodology[33]. Although the two approaches are not directly comparable, since PTAL incorporates additional information such as service frequency and walking times, both models aim at describing the accessibility of urban areas to public transportation. The comparison between Figure 5.4 and Figure 5.5 reveals a similar spatial distribution of accessibility levels across the city: in both cases, the highest accessibility values are concentrated in the central areas of Milan, where the transportation network is denser, while lower accessibility levels are mainly associated with peripheral neighbourhoods. Furthermore, both maps highlight several accessibility corridors corresponding to the main transportation infrastructures and some voids in terms of low accessibility can be found in both the representations. Even though the numerical values cannot be directly compared, the consistency between the two spatial patterns provides a first qualitative validation of the proposed model and suggests that the adaptive parametrisation introduced in this work is capable of capturing meaningful characteristics of Milan's transportation system.

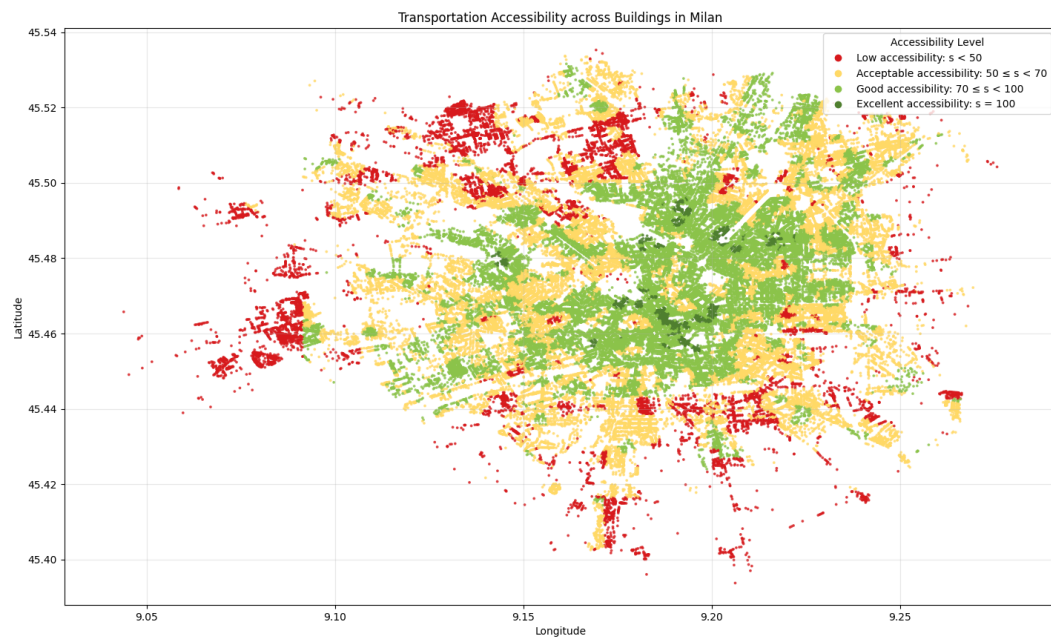


Figure 5.4: Map of Milan with parameterised thresholds.

30 min

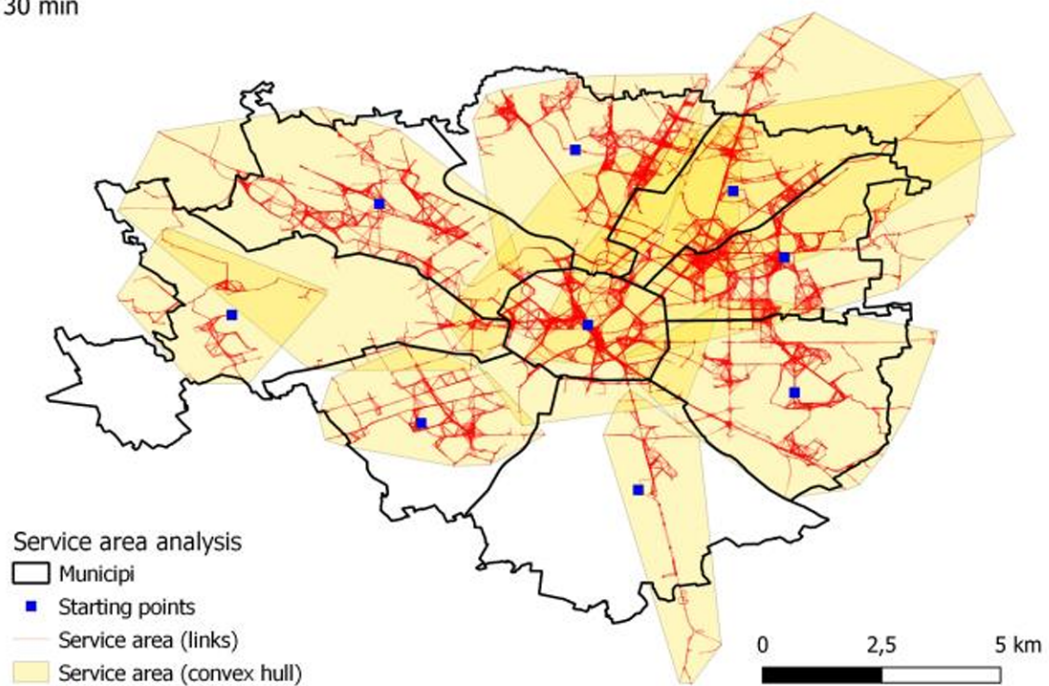


Figure 5.5: The 30 minutes accessibility model of Milan from PTAL paper.[33]

6 | A Context-Aware Representation to Support Urban Analysis

In this chapter, we will delve into a broader framework which aims at connecting all the various topics and methodologies contained both in the New European Bauhaus and in the CrAFt documents, introduced in the State of the Art. The tool containing all this information will finally create a contextual model which will further enrich the comprehension of the urban analysis and support the integration between the datasets and the indicators. As noted in the *Theoretical background* Section, in this chapter the method of the Context Dimension Tree will be adopted to create a tool which supports the analysis and to improve the quality (especially the accuracy) of the queries to the specific datasets. The necessity of introducing such a framework emerges from the heterogeneous nature of the information involved in urban analysis. Data collected from different sources often describe the same urban reality from different perspectives and levels of abstraction; consequently, establishing meaningful relationships among them becomes a fundamental step to extract knowledge rather than simply accumulate information. In this context, a structured representation of concepts and their connections can provide a common reference system capable of reducing ambiguities and facilitating the interpretation of complex datasets. Moreover, the indicators analysed throughout this thesis are seldom independent of one another: on the contrary, they are frequently connected through semantic relationships that reflect the intrinsic complexity of the urban environment itself. Understanding these relationships is therefore essential not only to improve the quality of the analysis, but also to ensure that different sources of information can contribute coherently to the same evaluation process. For these reasons, particular attention will be devoted to the definition of a hierarchical representation capable of organising concepts, attributes and entities according to their meaning and role within the broader urban context. Such an approach allows the analysis to move beyond the isolated observation of single variables and towards a more comprehensive interpretation of the information available. The

following sections will therefore introduce the conceptual structures adopted in this work and discuss their relevance within the overall methodological framework of the thesis.

6.1. The IMM Tree

Our first step is to build a framework that is coherent with the model provided by the CrAFt document (see paragraph 6.4). To do this, we adopted the *IMM hierarchical tree*, a conceptual model that provides a hierarchical structure with three different levels whose first level maps the 5 main intervention domains called *pillars*: the three main bottom-lines for sustainable development, which are *planet*, *people* and *prosperity*, complemented by a pillar on *quality of life* and one on *governance*. After the first level, we find a second level, which is composed of 17 impact categories referring to essential aspects of integrated sustainable development. The last level lists a series of relevant indicators, which have been identified based on a list of 46 indicators, intended as an indicative set of primary KPIs that are already largely used by most cities. To map this structure in python we used the *networkx* library, which provides the tools to build a graph-like structure where nodes need to be created giving them a name, and they need to be later connected with edges. In this particular case, it was sufficient to provide a hierarchical structure, thus the only type of relation name of the edges is the *hierarchy* relation. Once the graph was generated, a dedicated visualization function was implemented to display the resulting hierarchy. The layout was automatically computed through a top-down graph representation, allowing the different levels of the framework to be clearly distinguished and providing an immediate visual interpretation of the relationships among pillars, impact categories and indicators. Figure 6.1 shows a portion (or sub-tree) of the IMM hierarchical tree: in particular, this section is just an example to overview how the sections of the CrAFt framework have been mapped.

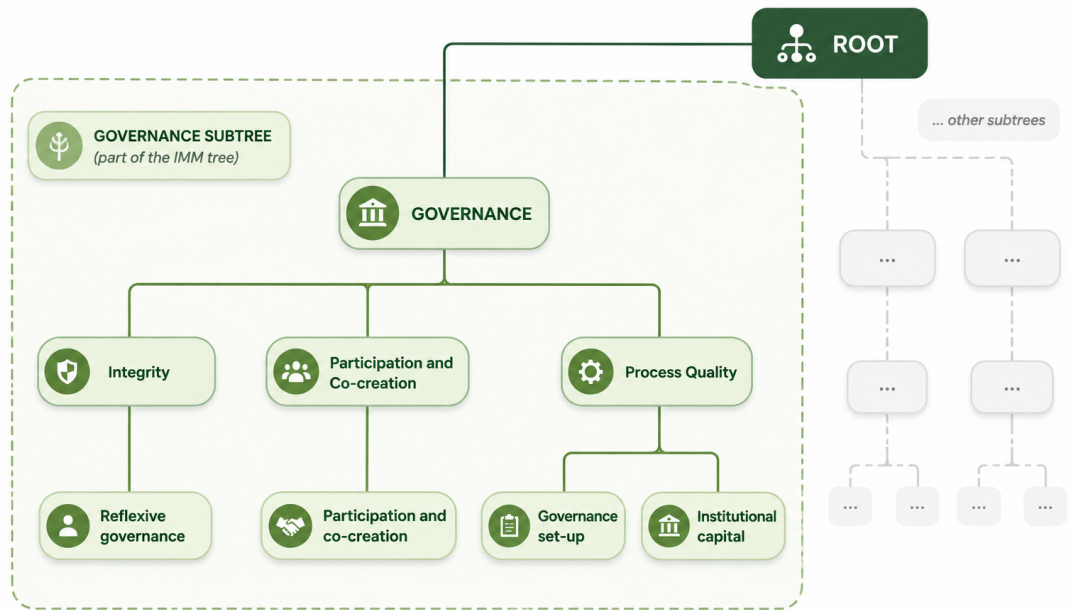


Figure 6.1: Governance sub-tree of the IMM tree.

6.2. The NEB Tree

The second framework to be represented is the one proposed by the New European Bauhaus initiative. Unlike the IMM structure, organised around intervention domains and impact categories, the NEB framework is built upon the three core values that define the initiative: *Sustainability*, *Beauty* and *Inclusivity*. These three dimensions constitute the first level of the hierarchy and act as the conceptual pillars through which urban transformation is evaluated. Each pillar is associated with a set of Key Performance Indicators (KPIs), identified through the official NEB documentation and intended to capture the different aspects of the three dimensions. As a result, the second level of the hierarchy is composed of the individual KPIs, each uniquely identified through a dedicated code. A KPI is then further decomposed into more specific sub-indicators or parameters, which provide a more detailed representation of the information required for its assessment. Consequently, the resulting structure is characterised by multiple levels of granularity, allowing both high-level concepts and detailed operational variables to coexist within the same framework. To implement this structure in python the *networkx* library was used, which provides the tools to build a graph-like structure where nodes need to be created giving them a name, and they need to be later connected with edges. Since some KPIs require a more detailed representation than others, the framework was designed to support

different levels of hierarchy, allowing specific indicators to be further decomposed into sub-indicators and operational parameters. This approach makes it possible to explicitly represent which information is required to evaluate a given KPI and how such information is related to the broader NEB dimensions. Finally, a visualization function was implemented to inspect the generated graph and verify the consistency of the hierarchy. In Figure 6.2 a portion of the NEB tree is represented: in particular, the reported subtree represents a part of the NEB tree which will be subsequently exploited to present the functionality of the Context Dimension Tree, and which contains the previously calculated accessibility KPI, the I.3.7 (see chapter 5).

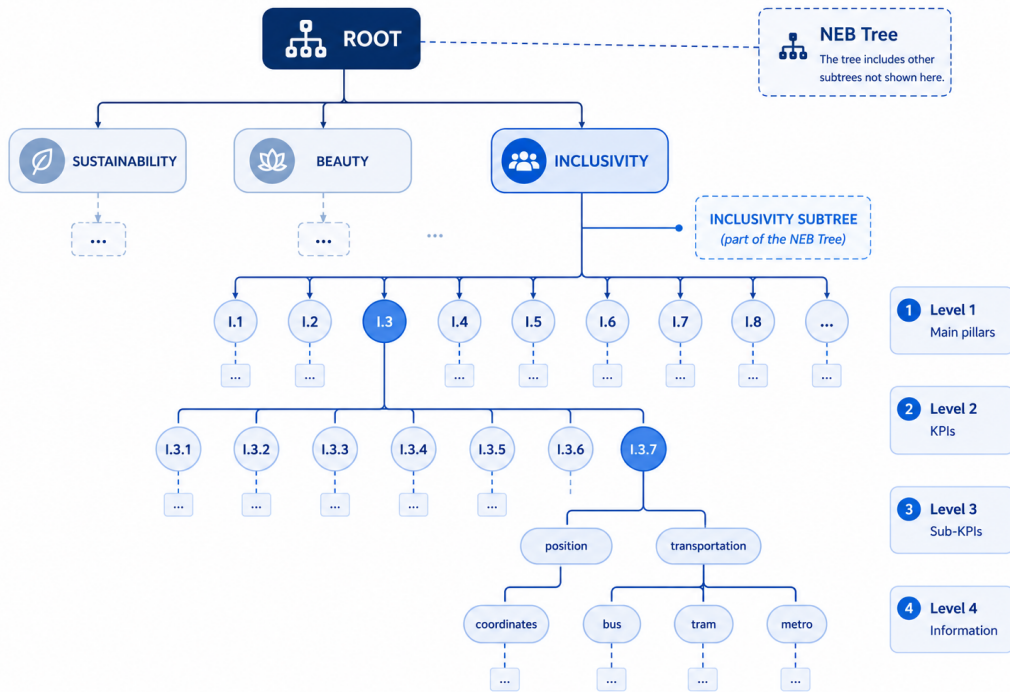


Figure 6.2: Sample from the NEB tree.

6.3. The Context Dimension Tree

In this section the application of the Context Dimension Tree (CDT) to this thesis is presented, together with its main elements and functions. To implement the context tree in Python, the NetworkX library was adopted. However, unlike the NEB and IMM trees introduced previously, which were designed as immutable structures a part of which is used here to work on, for each use case a CDT is produced as a conceptual model of all the possible contexts related to that use case. That is the reason why the CDT itself remains unchanged, but the visualisation changes according to the specific query: in

other words, only the contextual view corresponding to the required context is generated and represented, rather than the entire tree. This process is described in the following sections. Each node stores two additional pieces of information besides its name: its shape and colour. The theoretical background of the Context Dimension Tree has already been introduced in Chapter 3; therefore, only the aspects relevant to its implementation are discussed here: an example of the representation of the CDT developed in this thesis is reported in Figure 6.6.

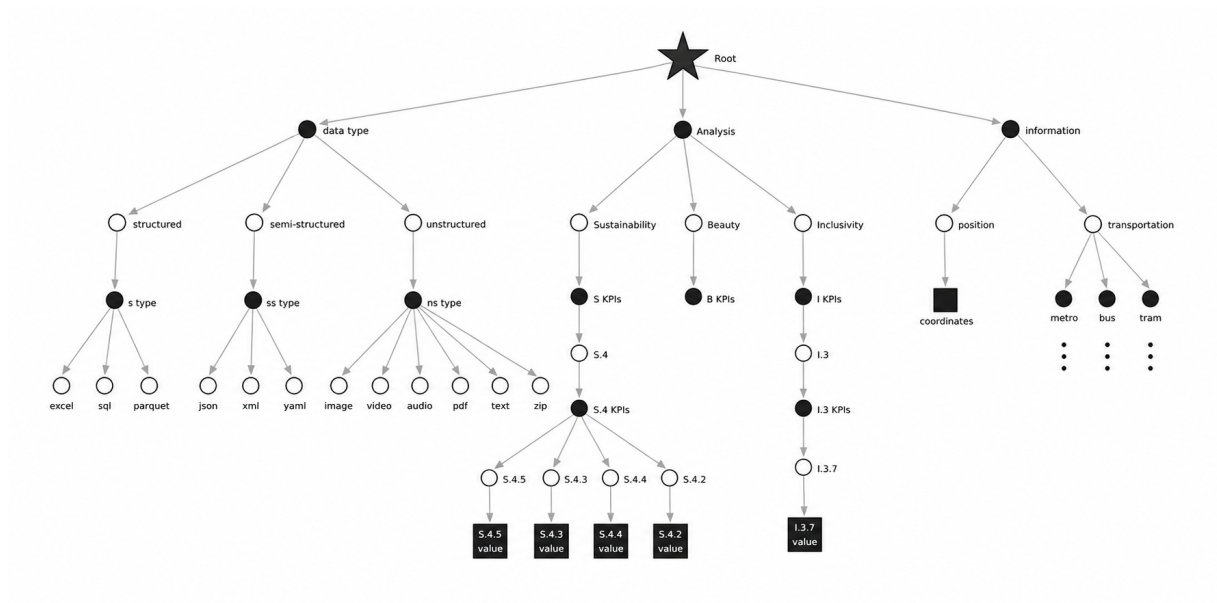


Figure 6.3: Example of the implemented CDT

6.3.1. Content of the implemented Context Dimension Tree

Once the general structure of the CDT is explained, we will concentrate on the actual content of our implemented tree. As described in literature, the first level after the root (star) is made of dimension nodes: in particular, this specific tree contains three dimension nodes, representing the different dimensions related to the urban analysis:

1. The first dimension is the **data type** dimension, which highlights the types of data inside of the given dataset, which are needed to calculate the indicators to support the analysis.
2. The second dimension is the **analysis** dimension, which is dedicated to the NEB indicators.
3. The third dimension is the **information** dimension, which lists the information related to the specific type of analysis.

These three dimensions were selected because they describe the three complementary aspects required to perform a context-aware analysis: the available data, the indicators to be evaluated and the information needed to compute them. Although all three dimensions belong to the same static Context Dimension Tree, they are populated differently according to their role within the framework. The data type dimension is defined during the initialisation of the context tree itself: this means that the nodes, which compose it, are statically initialized and cannot be changed in the following steps of the implemented pipeline. As mentioned above, this first dimension describes the types of data which can support the analysis and calculation of specific KPIs. The children of the dimension node are three concept nodes and they represent the value the data type can assume, which is *structured data*, *semi-structured data* and *unstructured data*. Additional nodes are attached to the latter ones and their nature is sub-dimensional: they can assume different values and in particular:

- Structured data can be of three different types: excel, sql, parquet.
- Semi-structured data can be: json, xml, yaml.
- Unstructured data can be: image, video, audio, pdf, text, zip.

The analysis dimension, on the other hand, is statically defined within the CDT, but only the portion of the corresponding sub-tree that is relevant to the current query is represented at runtime. This part will be explained in further details in the next sections. Similarly, only the relevant portion of the information sub-tree is represented once the information required by the current query has been identified. It is important to emphasise that this dynamic visualisation does not alter the theoretical structure of the Context Dimension Tree. The complete CDT still contains all the possible dimensions, concepts and contextual views defined during the modelling phase. The proposed implementation simply displays the subset of nodes that is relevant to the current query, providing a clearer and more readable representation of the context without modifying the underlying structure.

6.4. The Pipeline

In this section we present the overall pipeline of the creation of the context-aware view. Note that, while in this project almost every step of the pipeline has been implemented, there are still some technical limitations, such as the need for different types of data integration, both structured and unstructured, that require specific techniques and creates drawbacks from the implementation point of view. On the other hand, we have enough

material to exemplify the automatic nature of the whole pipeline, delivering a qualitative output which can be considered as an exemplary result.

The overall system is not completely automatic, since it must exchange data with a user: she or he is an actor who wants to analyse the urban context in order to make some decisions in terms of project feasibility or performance assessments.

The **context selection** extracts the ideal set of contextual information that would be required to analyse a given urban issue or query, and is obtained by identifying all the dimensions, concepts and indicators that are theoretically relevant to a specific context. Moreover, in the final step of the implementation, only some nodes of the context selection are highlighted (as shown in Figure 6.7), according to the actual availability of data, which is both the one stored in the system and the one provided by the user.

The steps of the implemented pipeline are described below and they are structured in a graphical pipeline, represented in Figure 6.4:

1. The first step is to give a specific input to the system: in particular, in this implementation's case the user must input a sentence (or text query) to the system, which describes the urban or architectural related intervention to be analysed through the NEB's directive.
2. Once the system has received the text query, it will elaborate it through the topics given by the CrAFt document and contained in the IMM tree.
3. After the most related impacts have been discovered, the system will recover the NEB's KPIs which best describe the given context.
4. Now all the necessary information has been gathered and a context selection will be created, returning to the user the main contextual dimensions: which indicators from the NEB best describe the context, which information is necessary to evaluate them and which types of data can actually deliver this kind of information.
5. At this point, if the contextual view needs to be additionally refined according to the availability of further data not present in the database (for example, the coordinates of a single point where the user wants the analysis to be carried out), these files need to be input by the user.
6. Once all the necessary data has been collected, the filtered contextual view is available, which is tailored according to the actual availability of the information.
7. Finally the results are displayed in the context tree and they can also be used to train a machine learning model.

In Figure 6.4, the overall pipeline has been drawn.

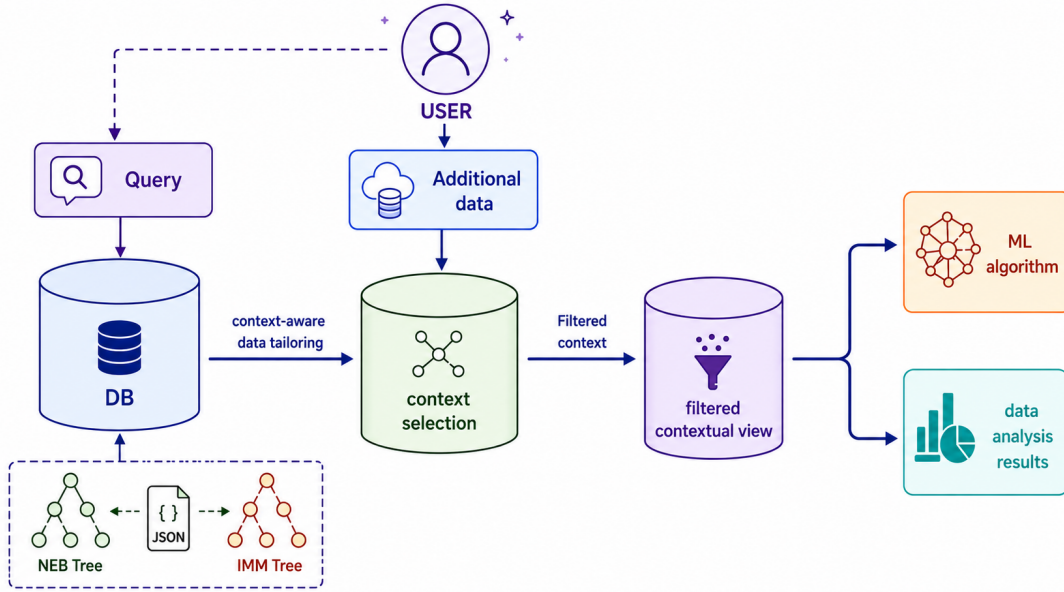


Figure 6.4: Pipeline for context-aware dataset creation.

To better understand the operation of the overall process, we give a concrete, detailed example in the next paragraph. This example is related to the I.3.7 KPI, presented in the previous Chapter.

Let us consider the problem of the construction of a new tram line in an urban context: the urbanist wants to know if the chosen path can be considered reasonable in terms of accessibility enhancement. The engagement with the context-aware system will start when the user inputs a sentence, for example, "New tram line to enhance public transportation". The system will then retrieve all the data from the NEB tree and the IMM tree: in particular, the sentence will be analyzed in terms of syntax and semantics, in order to find out the related topic(s) of the CrAFt document. In our case, the related topics are "Active and public transport" and "Diversity and accessibility of services" (in this example, we take into consideration just the two most compatible ones). At this point the system will retrieve the related NEB KPIs through a json file that acts as an interface between the NEB and IMM trees, together with the information needed to assess them: in this example, the ones related to public transportation. At this point the system will retrieve the related data from the system's database: this will contain structured, semi-structured, or unstructured data, already tagged and ordered and thus ready to deliver a first contextual view. In this case, the first contextual view will contain all the transportation-related data and indicators.

After the context selection is displayed, the user will feed additional data to the system, data that will be used to compute the various indicators, which will be tailored ad hoc. For example, if the user wants to assess the availability of a point of the new tram line, to understand if the path is coherent with the enhancement of transport accessibility, she or he will input the coordinates of the given point to the map: at this point the system will return a *filtered contextual view* which will support the analysis of the "New tram line to enhance public transportation" issue. The filtered contextual view can be subsequently analysed, either directly by the urbanist or through machine-learning algorithms. In the first case, the selected indicators provide an immediate overview of the accessibility conditions related to the proposed intervention; in the second case, predictive models can be employed to estimate future scenarios and identify possible criticalities. The process finally returns a set of results from the data-analysis, consisting of indicators and contextual information which support the evaluation of the validity of the new tram line and its contribution to improving public transportation accessibility.

Figure 6.5 shows how the system would work with the given example.

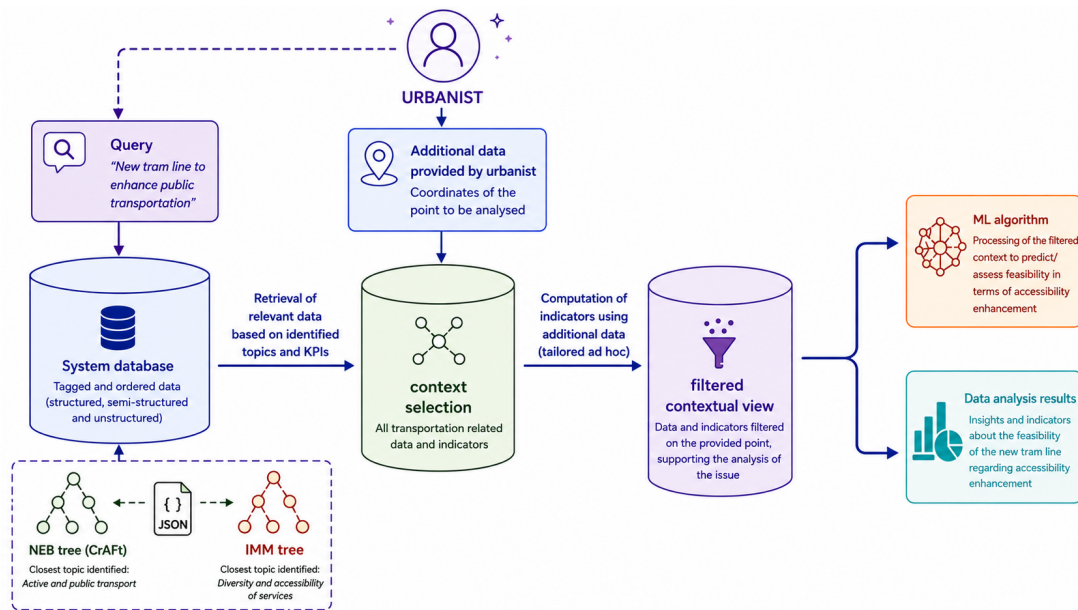


Figure 6.5: Example of the pipeline for context creation.

The following sections contain an overview on how specific aspects of the system's pipeline were implemented, from query input to the final filtered contextual view visualisation, together with brief references on the technologies that were used.

6.4.1. Elaboration of the Input

This section contains an overview on how the input sentence, given by the user, is elaborated in order to identify the most relevant topics contained in the IMM tree. Since the topics available in the CrAFt document are described through textual labels, the problem can be interpreted as a text-matching task, where the objective is to determine which impact categories are the most coherent with the intervention described by the user. To achieve this goal, the system combines both syntactic and semantic analysis techniques. The first one focuses on the similarity between the words contained in the sentence and the words describing the IMM topics; the second one attempts to capture the actual meaning of the sentence, even when different terms are used to express the same concept. The final score assigned to each topic is obtained by combining the results of both analyses, thus providing a more robust and reliable selection process. The implemented procedure can be summarised in the following steps:

- **Syntactic analysis.** The sentence provided by the user and the labels of all the leaf nodes contained in the IMM tree are transformed into a numerical representation through the TF-IDF (*Term Frequency-Inverse Document Frequency*) technique. This method assigns a weight to each term according to its relevance within the available corpus. Once the vectors have been generated, the cosine similarity is computed between the input sentence and every IMM topic. The resulting score quantifies the lexical similarity between the two texts, rewarding the presence of common words and expressions.
- **Semantic analysis.** A second evaluation is performed through a transformer-based language model. In particular, both the input sentence and the IMM topics are converted into semantic embeddings, namely numerical vectors capable of representing the meaning of a text. The cosine similarity is then computed between the embedding of the sentence and the embeddings of all the candidate topics. Unlike the previous approach, this technique is capable of identifying conceptual similarities even when different words are used. For example, terms such as *public transportation*, *mobility services* and *urban transit* can be recognised as semantically related despite presenting limited lexical overlap.

The final ranking is obtained by combining the syntactic and semantic scores through a weighted average, assigning a greater importance to the semantic component. The highest ranked topics are subsequently selected and used to navigate the IMM tree, retrieve the corresponding NEB indicators and generate the contextual view that will support the following phases of the analysis.

6.4.2. CDT Dimensions Selection

Once the topics from the IMM tree have been ranked, the correspondent KPIs from the NEB tree must be found. The New European Bauhaus, as mentioned before, lists a series of main indicators covering the three main pillars of the directive, but these indicators are calculated from other sub-indicators. A singular KPI can be obtained from very diverse sub-indicators, which could appear to have nothing in common between them; this is the reason why the connection between the CrAft document and the NEB method was carried out through the low-level KPIs and not through the main, high-level ones.

A fitting example can better explain this concept: if we take the "Active and public transportation" topic from the CrAft, we find sustainability main indicators which deal with it (e.g., the S.4 – Sustainable Mobility), but only some of the indicators that compose it are related to public transportation (e.g.: S.4.1 is E-Mobility, which does not belong to active or public transportation). Moreover, the connection between CrAft topics and NEB indicators cannot be made using semantic or syntactic similarities, since this kind of operation of the overall pipeline is considered vital for delivering effective results. So, this part of the implementation must be done manually. In particular, a json file must be created: all the IMM-CrAft topics must be written inside and connected to a list of inherent sub-indicators from the New European Bauhaus. The information contained in the json file are read by the context tree function, which generates the context selection on that specific knowledge.

6.4.3. CDT Nodes Selection

As mentioned in the previous chapters, the implemented visualization of the context tree differs from the one contained in the theoretical representation: this is due to return a tidier and clearer version of the contextual view, since all the concept and dimension nodes are too many to be reported as a whole. Once the sub-KPIs from the json file have been retrieved, the system must interact with the NEB tree to collect all the information needed to compute those specific indicators. Once given the name of the sub-indicator as parameter, a function retrieves the sub-tree made of all the different pieces of information to calculate it. The output of the function is then passed as parameters to a specific function of the context tree which has a double functionality: it both adds the KPI and the sub-KPI to the tree, under one of the analysis concepts (Sustainability, Beauty and Inclusivity) and it also adds the related information to the Information node. The previous sections of this chapter explained how the sub-trees of the CDT populate themselves, but how does this translate into the actual functional implementation? In other words, how

do the three main dimensions of the tree interact with each other? For instance, there is a clear connection between the Analysis dimension and the Information dimension: this is due to the design of the NEB tree, where the KPIs nodes are directly connected to the information needed to calculate them. This makes it easy to assess which contextual items contribute to the creation of the context through the KPIs, since it is sufficient to iterate on the NEB tree to return the coupling between the dimensions. The challenging part of this process is to understand which kinds of files can actually support the analysis by delivering the needed information. This aspect was done manually in this project, for a matter of simplicity and it was implemented for just a few specific cases. Theoretically, all types of data can deliver a particular piece of information: for example, the coordinates of a position can be extrapolated from a csv file, from an image (both as metadata or from the representation of the image itself), from an audio, et cetera. In this project's implementation, it was decided to narrow down the range of options and to force the recognition of the needed information only from a few types of data. This step of the pipeline delivers a first contextual view given from the input issue, which answers three different questions:

- *Data type dimension*: Which data types deliver the information to describe the context?
- *Analysis dimension*: Which indicators from the NEB framework are fit to analyse the context?
- *Information dimension*: Which information IS needed to compute the NEB indicators?

Figure 6.6 represents an example of the contextual view given by the input issue "*Enhancement of public transportation services*":

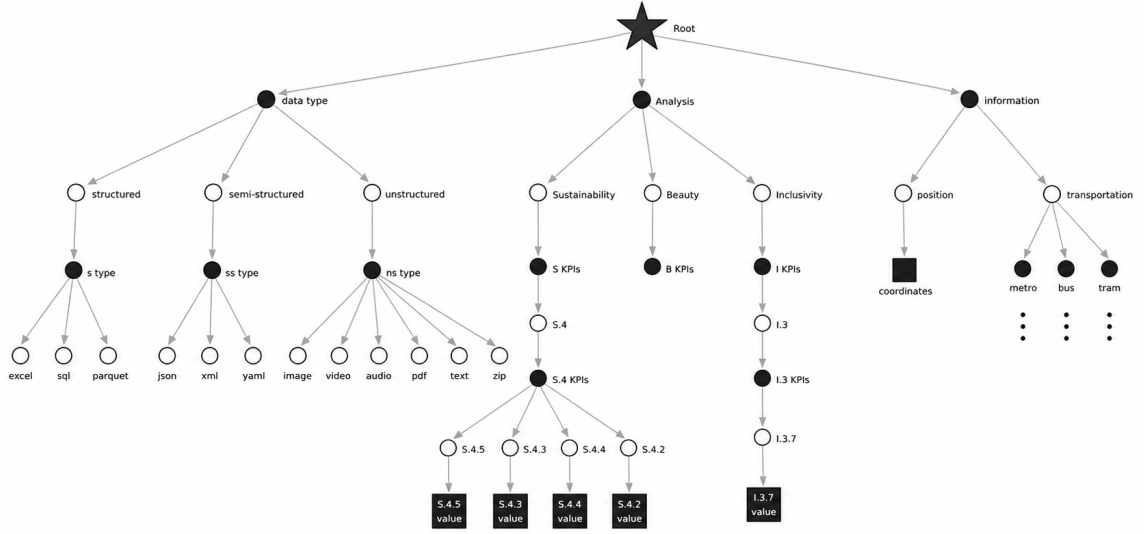


Figure 6.6: Example of Context Tree.

Once the context tree has been initialised as described above, it is time to highlight the contextual nodes according to the data which is actually available to process. At this point, a second input must be given by the user to the system, in particular two different types of inputs are asked: one is the additional data which is not present in the system's DB and which is required to compute the KPIs and the other is the relative data, which filters the contextual view. An example of the two inputs is made to better explain the theoretical assumptions made above. If we consider the previously made example of the *"New tram line"*, we might uncover a few scenarios where it is necessary to provide additional information to the ones already present in the system's database. The aim is to assess if the chosen path of the tram can enhance accessibility to public transportation: the I.3.7 KPI and from the S.4.3 to the S.4.5 are the perfect indicators to analyse that. Imagine we only have the information regarding the tram network inside the database, which comprises the number of the lines and coordinates of the various stops. The user is able to input two additional files: one of them is a csv file which contains the metro network, with its lines and the coordinates of the stops (**additional data**) and the other is a photo of the street where the municipality approved the passage of the tram (**relative data**). This way, some KPIs can be computed, but they refer to only some information (in this case they do not refer to the bus network) and some cannot be computed at all (in this example it is impossible to calculate S.4.4, since no data about transportation usage is available). Finally, the system returns a **filtered contextual view**, which is represented in Figure 6.7.

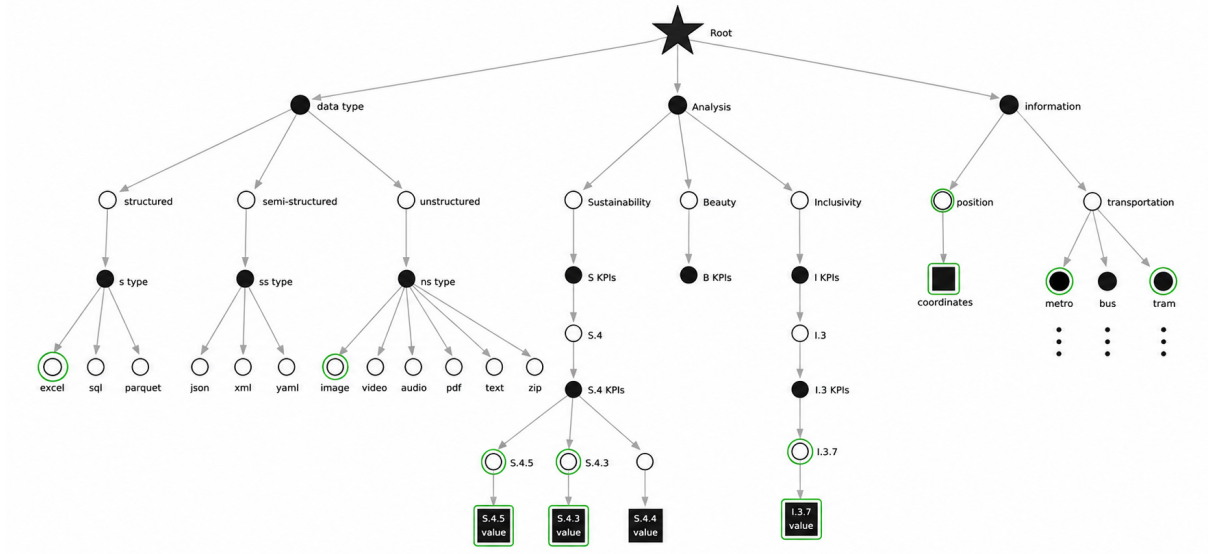


Figure 6.7: Example of Filtered Context Tree.

To summarise, the system is available to return a filtered contextual view, where the context of the specific urban analysis is well highlighted through both the topic and the available information in the form of datasets. So, to describe the context with the three different questions, which mirror the three main dimensions of the context tree, we define the context as follows:

- Which data types are present in the system?
- Which KPIs can be computed with the provided information?
- Which information was extracted from the input files?

Overall, the Context Dimension Tree represents a fitting tool to support the production of the final outcome of the proposed framework. It integrates the information extracted from the IMM and NEB trees into a context-aware representation that supports the analysis, in this case, of urban issues, according to the available data.

7 | Future Developments

The work presented in this thesis demonstrates how heterogeneous urban data can be analyzed through the integration of the New European Bauhaus framework, the CrAft Impact Model and the Context Dimension Tree (CDT) representation. Although the proposed pipeline is capable of generating contextual views while supporting the selection of indicators according to a specific urban issue, several components of the system still present opportunities for further development. Some parts of the implementation were intentionally simplified, in order to validate the overall methodology while focusing on the construction of the contextual framework. In particular, the current version of the system relies on manually defined relationships among impact categories, indicators and information sources, and requires a preliminary organization of the available datasets before they can be effectively integrated into the contextual analysis. This Section provides an overview of the possible evolutions of the framework.

7.1. The Impact Model

One of the possible future developments concerns the extension of the current implementation towards a dynamic representation of the CrAft Impact Model. In the present version of the framework, the IMM tree is mainly exploited as a hierarchical structure capable of connecting user issues to the corresponding impact categories and, consequently, to the most relevant NEB indicators. However, the CrAft document proposes a much richer perspective, where urban interventions are not analyzed as isolated actions but as elements of a network of interconnected impacts. In particular, the implementation already includes a set of non-hierarchical relationships between impact categories, which describe how a topic can influence another one through three different mechanisms:

- *improves*, representing a positive impact;
- *reduces*, representing a negative impact;
- *enables*, representing a necessary condition for another effect to occur.

Currently, these relationships are presented inside the IMM graph structure but are not

actively exploited during the analysis process. A possible evolution of this framework would consist in transforming these connections into a simulation environment capable of modelling the consequences of urban interventions. Instead of using the IMM solely as a topic selection mechanism, the graph could be interpreted as a network of relationships where the changes in one impact category propagate through the connected nodes. For example, consider the issue *Enhancement of public transportation services*. After the most relevant topics have been retrieved from the Impact Model, the system identifies *Active and public transport*, *Location and space use*, and *Improved accessibility of services* as the topics most strongly related to the user's request. The selected topics do not represent isolated concepts, though: according to the relationships defined inside the Impact Model, they are connected to several other impact categories through *improves*, *reduces* and *enables* relationships. As a consequence, the enhancement of public transportation services can potentially generate effects on multiple dimensions of the urban environment. The possible impacts are reported in Figure 7.1.

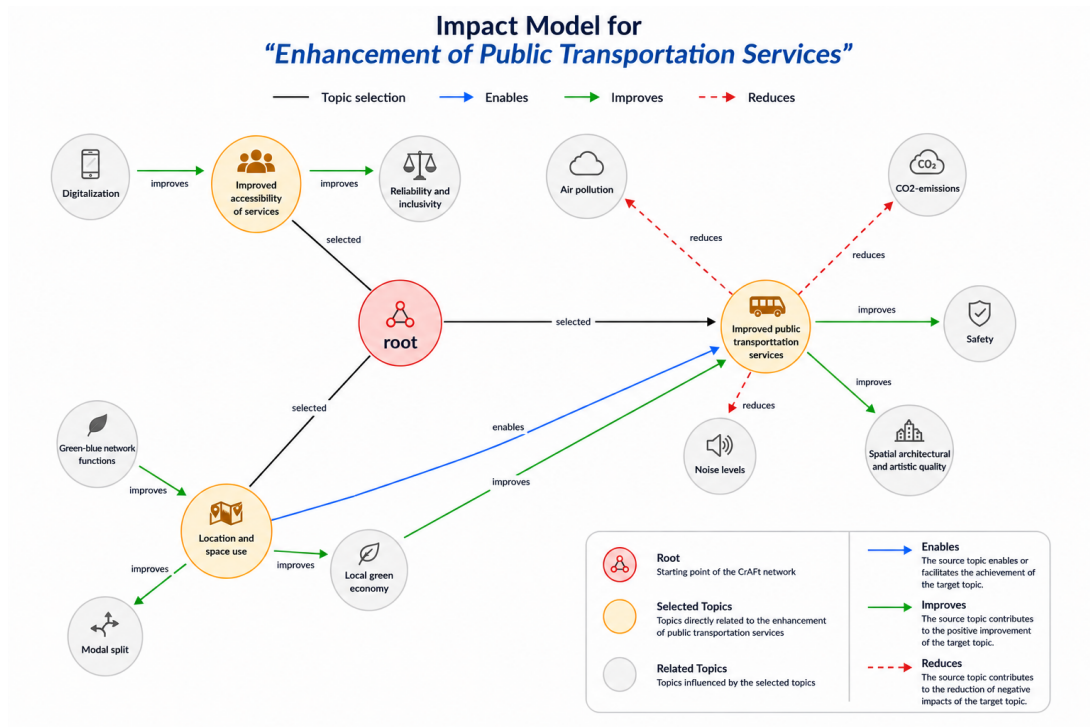


Figure 7.1: Example of Impact Model graph.

The possibility to model these interactions would transform the framework from a contextual analysis tool into a decision-support system. Urban planners could simulate alternative interventions and observe how variations in a specific indicator propagate across different dimensions of the urban environment. Moreover, the same approach could be

extended to the New European Bauhaus indicators. Once the relations between CrAft topics and NEB KPIs have been established, modifications applied to one indicator could generate estimations on other related indicators, providing a more comprehensive understanding of the expected outcomes of an intervention. Such an extension would be coherent with the original objectives of the CrAft Impact Model, which was conceived as a whole-system approach capable of highlighting co-benefits, trade-offs and interactions between different dimensions of urban sustainability.

7.2. Automatic Integration of User-Provided Datasets

As described in the previous chapter, the creation of the first contextual view only depends on the issue provided by the user. Once the most relevant CrAft topics have been identified and the corresponding NEB indicators have been retrieved, the Context Tree is capable of highlighting which information is needed to assess the selected KPIs. However, this first contextual view only represents a 'potential' context: at this stage, the system still does not know which information are actually available to perform the analysis. For this reason, a second input is required from the user. In particular, the user must provide the datasets that should be used to compute the indicators and support the analysis. In the current implementation, this process is partially manual: each dataset was inspected before and its content was explicitly connected to the corresponding information nodes of the Context Tree. As a consequence, the system already knows which information can be extracted from a specific file and how they contribute to the generation of the filtered contextual view.

To better understand this limitation, we should consider again the transportation example discussed throughout this chapter. Suppose that the user uploads a CSV file containing the position of the metro stations of a city. In the current implementation, the system is able to retrieve the coordinates contained in the file because this association has already been manually defined during the development phase. However, if a different dataset was provided, for example a CSV containing tram stops or a JSON file describing public transportation routes, the same operation would require a new manual inspection of the dataset before it could be integrated into the analysis. A possible future development of the framework would therefore consist in automating this phase of the pipeline. Instead of relying on manually defined associations, the system could analyse the uploaded datasets and automatically identify the information contained within them. The process could be summarised through three main steps:

1. identification of the structure of the dataset;

2. extraction of the meaning of the available attributes;
3. integration of the extracted information into the Context Tree.

The first step would depend on the type of data provided by the user. Structured datasets such as CSV, SQL or EXCEL files could be analysed through their schema and column names; semi-structured formats such as JSON, XML and YAML files could be explored through their hierarchical structure; unstructured data would instead require dedicated extraction techniques depending on the media type.

Once the structure of the dataset has been analysed, the system should understand which information is actually represented inside the file. For example, a column called *longitude*, *lng* or *x-coordinate* could be recognised as geographical information, while attributes such as *buildingheight*, *floors* or *elevation* could be associated with urban morphology descriptors. The same principle could be applied to transportation datasets, environmental measurements or demographic information.

This idea is coherent with several works proposed in the literature: PerLa [28], for example, introduces an intermediate abstraction layer capable of hiding the heterogeneity of the underlying data sources and providing a common representation for data collection and processing activities. In a similar way, the Information dimension of the Context Tree could act as an abstraction layer between the available datasets and the indicators that require this information.

A second relevant aspect concerns the semantic interpretation of the extracted data. Recent works on semantic data preparation suggest enriching datasets with domain knowledge before performing integration and analysis tasks. According to the S-PIC4CHU [34] project, semantic information can support data integration, transformation and quality management by explicitly representing the meaning of the available data rather than only their syntactic structure. A similar concept can also be found in multimedia retrieval systems such as SHIATSU [35] and Scenique [36]. In both cases, raw multimedia content is associated with semantic concepts organised through hierarchical and multidimensional structures, allowing the system to reason on meanings rather than on low-level features alone. Although the application domain is completely different, the underlying principle remains the same: transforming heterogeneous data into semantic concepts that can subsequently be exploited by higher-level processes.

In the context of this thesis, these concepts could be used to automatically populate the Information dimension of the CDT. Once the required information has been identified inside the uploaded datasets, the corresponding nodes could be highlighted and connected

to the contextual view. The resulting filtered context would therefore no longer depend on manually predefined associations, but on the actual content of the data provided by the user. This extension would considerably improve the flexibility of the framework. New datasets could be integrated without requiring dedicated implementations, while the generation of the filtered contextual view would become a fully dynamic process. Ultimately, the Context Dimension Tree would not only identify which information is needed to analyse a specific urban issue, but it would also be capable of automatically understanding which information is available and how it contributes to the selected indicators.

8 | Conclusions

This thesis started from a specific research problem: the New European Bauhaus provides a rich and ambitious framework to evaluate urban environments, but many of its indicators are still difficult to compute automatically and, above all, they often require a contextual interpretation in order to become actually meaningful. In this perspective, the work presented in this thesis tried to connect automatic data processing, indicator computation and context representation within a single methodological framework.

The first contribution of the thesis concerns the automation of the computation of two existing Beauty indicators, namely B.7 and B.11. Starting from a previous multimedia pipeline, the work focused on replacing manual operations with automatic procedures based on Computer Vision techniques. In particular, SAM3 was used to support facade rectification and architectural element recognition, while YOLO was tested as a possible strategy to improve the scalability of the detection process. Although some limitations remain, especially in terms of computational efficiency and precision for less represented architectural elements, the proposed pipeline shows that the extraction of visual information from facade images can be significantly automated, reducing the amount of manual work required to compute aesthetic indicators.

The second contribution is related to the implementation of the I.3.7 indicator, which evaluates accessibility to public transportation. Unlike the Beauty indicators, this KPI required a different type of analysis: the image alone was no longer sufficient, since the assessment depended on the position of the building and on the transportation network around it. For this reason, metadata extracted from images were combined with structured mobility datasets, allowing the indicator to be computed automatically for buildings located in Milan. Moreover, the original thresholds proposed by the NEB were adapted to the specific characteristics of the Milanese context: this made it possible to obtain a more meaningful evaluation of accessibility, avoiding an excessively generic interpretation of the indicator.

The third and most important contribution of the thesis is the adoption of the Context Dimension Tree as a conceptual and computational framework for context-aware urban

analysis. The CDT made it possible to organise the information extracted from the previous steps and to connect it with the NEB indicators, the CrAFt Impact Model and the available datasets. The framework does not only compute indicators, but also clarifies the gap between the analytical objective and the real data conditions supporting the analysis. In particular, a context selection is generated according to a given query, and a filtered contextual view will be subsequently displayed, which highlights not only the contextual view of the given issue, but also what can be computed with the available data.

Overall, the proposed methodology demonstrates that the computation of NEB indicators can be progressively transformed from a set of isolated procedures into a broader context-aware framework. The automatic computation of B.7, B.11 and I.3.7 provides the first operational layer of the system, while the CDT offers a structured representation capable of organising the results and connecting them to heterogeneous contextual information. This integration represents a first step towards a more flexible tool for urban analysis, where indicators are not interpreted as independent numerical values, but as part of a wider network of data, meanings and possible urban contexts.

At the same time, the work also highlights several limitations. Some parts of the pipeline are still partially manual, especially the connection between CrAFt topics, NEB indicators and required information; moreover, the integration of user-provided datasets still depends on predefined assumptions about their structure and semantic content. These aspects do not invalidate the proposed framework, but rather define its current boundaries: the thesis demonstrates the feasibility of the approach and provides a clear direction for future developments, especially regarding the automatic interpretation of heterogeneous datasets and the dynamic exploitation of the relationships contained in the Impact Model.

In conclusion, this thesis shows how Computer Vision, urban data analysis and context representation can cooperate to support the interpretation of New European Bauhaus indicators. The final result is not only a set of implemented KPIs, but a framework that tries to make their use more coherent with the complexity of the urban environment: data are extracted, indicators are computed, and context is represented in order to support more informed and context-aware urban analyses.

Bibliography

- [1] Eugenio Cesario. Big data analytics and smart cities: applications, challenges, and opportunities. *Frontiers in Big Data*, Volume 6 - 2023, 2023.
- [2] Ylenia Casali, Nazli Yonca Aydin, and Tina Comes. Machine learning for spatial analyses in urban areas: a scoping review. *Sustainable Cities and Society*, 85:104050, 2022.
- [3] Pierfrancesco Bellini, Monica Benigni, Riccardo Billero, Paolo Nesi, and Nadia Rauch. Km4city ontology building vs data harvesting and cleaning for smart-city services. *Journal of Visual Languages Computing*, 25(6):827–839, 2014. Distributed Multimedia Systems DMS2014 Part I.
- [4] David Victor Ogunkan and Stella Kehinde Ogunkan. Exploring big data applications in sustainable urban infrastructure: A review. *Urban Governance*, 5(1):54–68, 2025.
- [5] Dapeng Liu and Victoria Y. Yoon. Developing a goal-driven data integration framework for effective data analytics. *Decision Support Systems*, 180:114197, 2024.
- [6] Forensic Architecture. Forensic architecture, 2026. Accessed June 22, 2026.
- [7] Forensic Architecture. A cartography of genocide: Israel’s conduct in gaza since october 2023, 2024. Investigation published by Forensic Architecture.
- [8] Špela Verovšek and Miha Moškon. Context-specific urban optimisations through data-driven classification: A perspective on methods and applications. *Land*, 14(8):1505, 2025.
- [9] Konstantinos Gkatzogiannis, Romano Romano, and Paola Negro, editors. *A Practical Guide to the New European Bauhaus Self-Assessment Method and Tool*. European Commission, Joint Research Centre, 2024.
- [10] Han Vandevyvere and Annemie Wyckmans. Neb impact model: Summary for cities. Project deliverable d1.1, CrAFt – Creating Actionable Futures, June 2023. Version 30.06.2023.

- [11] Cristiana Bolchini, Carlo A. Curino, Giorgio Orsi, Elisa Quintarelli, Rosalba Rossato, Fabio A. Schreiber, and Letizia Tanca. And what can context do for data? *Communications of the ACM*, 52(11):136–140, 2009.
- [12] Carlo Ratti. Strategic vision of the building engineering of the future. Course slides, Politecnico di Milano, 2026. Architecture MSc course.
- [13] European Commission. New european bauhaus, 2026.
- [14] Ahmet Gun. Urban design evolved: The impact of computational tools and data-driven approaches on urban design practices and civic participation. *Journal of Contemporary Urban Affairs*, 7(1):242–260, 6 2023.
- [15] Ahmed A. Zakaria, Tasneem Amr, and Amany A. Ragheb. Iot in smart urban planning: A comprehensive review of applications, developments, and engineering perspectives. *IEEE Access*, 13:135316–135335, 2025.
- [16] Comune di Milano. Il gemello digitale del comune di milano, 2025. Accessed June 22, 2026.
- [17] European Space Agency. Sar interferometry shows edges of venice in motion, 2001. Accessed June 22, 2026.
- [18] Davide Ciuffreda and Marco Ciammaichella. Multimedia data analysis: A pipeline for the new european bauhaus kpis. Master’s thesis, Politecnico di Milano, 2026. Master’s Degree in Computer Science and Engineering.
- [19] Tommaso Andaloro, Ilaria Bartolini, Carlo Andrea Biraghi, Marco Ciammaichella, Davide Ciuffreda, Emilia Lenzi, Marco Patella, and Letizia Tanca. A pipeline for multimedia data preparation in the context of the new european bauhaus. In *Proceedings of the 34th Symposium on Advanced Database Systems (SEBD 2026)*, Senigallia, Italy, June 2026. Discussion Paper.
- [20] Sagar Joglekar, Daniele Quercia, Miriam Redi, Luca Maria Aiello, Tobias Kauer, and Nishanth Sastry. Facelift: a transparent deep learning framework to beautify urban scenes. *Royal Society Open Science*, 7(2):190987, 2020.
- [21] Xinyu Liang, Kaixiang Zhuang, Yun Wang, Yueting Su, Jianfeng Feng, Feng Zhou, Benjamin Becker, and Deniz Vatansever. Latent neural architecture organising shared aesthetic evaluations of visual artworks. *Nature Communications*, 2026. Article in press.
- [22] Yiqun Chen, Soheil Sabri, Abbas Rajabifard, and Muiyiwa Elijah Agunbiade. An

- ontology-based spatial data harmonisation for urban analytics. *Computers, Environment and Urban Systems*, 72:177–190, 2018.
- [23] R. Marasinghe et al. Computer vision applications in urban planning: A systematic review. *Sustainable Cities and Society*, 100:105047, 2024.
- [24] Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, Jie Lei, Tengyu Ma, Baishan Guo, Arpit Kalla, Markus Marks, Joseph Greer, Meng Wang, Peize Sun, Roman Rädle, Triantafyllos Afouras, Effrosyni Mavroudi, Katherine Xu, Tsung-Han Wu, Yu Zhou, Liliane Momeni, Rishi Hazra, Shuangrui Ding, Sagar Vaze, Francois Porcher, Feng Li, Siyuan Li, Aishwarya Kamath, Ho Kei Cheng, Piotr Dollár, Nikhila Ravi, Kate Saenko, Pengchuan Zhang, and Christoph Feichtenhofer. Sam 3: Segment anything with concepts. *arXiv preprint arXiv:2511.16719*, 2026.
- [25] Jian Wei, Qinzhaoh Wang, and Zixu Zhao. Yolo-g: Improved yolo for cross-domain object detection. *PLOS ONE*, 18(9):e0291241, 2023.
- [26] Ultralytics. K-fold cross validation with ultralytics, 2023. Documentation, last updated 2025.
- [27] Fabio A. Schreiber, Letizia Tanca, Romolo Camplani, and Diego Viganò. Managing and using context information within the perla language. Lecture slides, Politecnico di Milano, Dipartimento di Elettronica e Informazione, 2012. Course material on Context Dimension Trees (CDT) and PerLa context management.
- [28] Fabio A. Schreiber, Romolo Camplani, Marco Fortunato, Marco Marelli, and Guido Rota. Perla: A language and middleware architecture for data management and integration in pervasive information systems. *IEEE Transactions on Software Engineering*, 38(2):478–496, 2012.
- [29] Roboflow. Roboflow. <https://roboflow.com/>, 2026. Accessed: 25 June 2026.
- [30] Comune di Milano. Portale open data del comune di milano. <https://dati.comune.milano.it/web/portale-del-dato>, 2026. Accessed: 24 June 2026.
- [31] Comune di Milano. Atm - composizione percorsi linee metropolitane. https://dati.comune.milano.it/dataset/ds533_atm-composizione-percorsi-linee-metropolitane, 2026. Accessed: 24 June 2026.
- [32] Comune di Milano. Atm - composizione percorsi linee di su-

perficie urbana. <https://dati.comune.milano.it/dataset/ds532-atm-composizione-percorsi-linee-di-superficie-urbane>, 2026. Accessed: 24 June 2026.

- [33] Gianmarco Naro, Carlo Andrea Biraghi, and Emilia Lenzi. City transport analyzer: A powerful qgis plugin for public transport accessibility and intermodality analysis. In *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume XLVIII-4/W12-2024, pages 113–119. Copernicus Publications, 2024.
- [34] Gianvincenzo Alfano, Ilaria Bartolini, Diego Calvanese, Paolo Ciaccia, Sergio Greco, Davide Lanti, Emilia Lenzi, Davide Martinenghi, Cristian Molinaro, Marco Patella, Letizia Tanca, Riccardo Torlone, and Irina Trubitsyna. S-PIC4CHU: Semantics-based provenance, integrity, and curation for consistent, high-quality, and unbiased data science. In *Proceedings of the 33rd Symposium on Advanced Database Systems (SEBD 2025)*, CEUR Workshop Proceedings, Ischia, Italy, June 2025.
- [35] Ilaria Bartolini, Marco Patella, and Corrado Romani. Shiatsu: Tagging and retrieving videos without worries. *Multimedia Tools and Applications*, 63(2):357–385, 2013.
- [36] Ilaria Bartolini and Paolo Ciaccia. Scenique: A multimodal image retrieval interface. In *Proceedings of the International Working Conference on Advanced Visual Interfaces (AVI 2008)*, pages 476–477, Naples, Italy, 2008. ACM.

List of Figures

1.1	Preview of <i>A cartography of genocide</i> article.	3
1.2	Overview of the thesis' work.	5
2.1	Milan's digital twin.	10
2.2	The New European Bauhaus logo and promotional poster.	12
2.3	Image taken from NEB official website.	13
2.4	The CrAFt framework.	15
2.5	The pipeline of the previous thesis.	16
2.6	The Facelift framework.	18
2.7	Research gaps and aim of the thesis framework.	21
3.1	Example of building segmentation in an urban landscape.	26
3.2	YOLO's structure framework.	27
3.3	Cross validation process.	28
3.4	Example of Context Tree from the PerLa system [28], with two possible contexts from this tree.	31
4.1	Examples of images from the dataset.	34
4.2	Facade rectangularisation pipeline.	36
4.3	Architectural element recognition refined.	40
4.4	YOLO cross-validation pipeline.	42
4.5	Normalised confusion matrix and precision recall matrix with 10 epochs.	45
4.6	Normalised confusion matrix and precision recall matrix with 50 epochs.	45
4.7	Examples of bounding boxes processed by cross-validation.	46
4.8	Example of the symmetry implementation pipeline.	48
5.1	Original KPI thresholds from the NEB.	52
5.2	Original KPI classes and thresholds from the NEB.	55
5.3	Representation of Milan's buildings' accessibility with the NEB original thresholds.	56
5.4	Map of Milan with parameterised thresholds.	59

5.5	The 30 minutes accessibility model of Milan from PTAL paper.[33]	59
6.1	Governance sub-tree of the IMM tree.	63
6.2	Sample from the NEB tree.	64
6.3	Example of the implemented CDT	65
6.4	Pipeline for context-aware dataset creation.	68
6.5	Example of the pipeline for context creation.	69
6.6	Example of Context Tree.	73
6.7	Example of Filtered Context Tree.	74
7.1	Example of Impact Model graph.	76